**5/12/2019**

Had this great idea to practice the data structures that I learned in my CSC 300 class. So far this sounds like a good and bad idea.

**5/13/2019**

Started to a large bunch of the programming for the doubly linked lists today. I plan on making a doubly circular linked list. Currently using C++ as the language to write my data structures in. It’s 10 pm and I’m more concerned about watching Spiderman the animated series instead of figuring out the errors that I have in my code.

**5/17/2019**

So, I currently have solved all but three of the errors in my linked lists header files. Currently while programming I'm watching Batman begins. I have no idea what to write in this is log, I have no idea why I have one for these practice assignments.

**5/23/19**

Its been almost a week since I last worked on this little project of mine today. Solved the last of the errors today in my node and linked list file. First, I had an error where I was pointing in a file pointer into a normal integer array, this was happening for my setNext and setPrev functions. A lot of the other errors that I solved on 5/17 entry I forgot what they were, so I didn’t can’t really record what I changed. Changed print list to be a for loop that starts at one and continues until the loop count variable is equal to the number of nodes created in the list. Updated print list function again to have the function be able to traverse backwards if the next pointer is equal to null. Updated the add middle function where the value added to the list and the location where its going to take place at needs to be added into the list. Adds middle works of the head pointer getNext is null that way it can be added in from working in backwards from the list. It’s close to 11:50 p.m. going to comment my code then I’m going to push it to github and go to bed afterwards.

**5/24/19**

When I went to bed early this morning I realized a small problem with my program, I made a singular linked list where the only true doubling part comes in the addMiddle function; not only that, but I missed the whole part of the circular part of the list as well – fixed that by setting the previous head and tail pointer to point to the new object before head or tail becomes equal to that new object. Changes are successful, had to change the clear function so that it doesn’t continue while the walker node is null, but instead it goes by using the count variable to measure the number of nodes that need to be cleared out from memory. Deleted the code in else statement in the addMiddle function because it became useless to have since it was based off the mistake that I mentioned in the begging of this log entry. I believe that I finished the doubly circular linked list project that I started off on a while ago, going to moving onto stack and queues pretty soon.

**5/25/19**

Started stacks and queues today, think about using the Node.hpp file that I used for my doubly linked lists just, so I don’t have to rewrite a bunch of code. Decides to make a separate class for stacks and a separate class for queues to keep everything separated and easier to work with.

**6/4/2019**

Decided to split the stacks and queues into their class to help simplify the program progress. Added in the addStack, seeStack, removeHead, and clear functions. Had really no errors as of now, except for a small part where I accidently wrote void Stack<T>::Stack() instead of Stack<T>Stack(). Finished the stack class at the moment, pretty confident that the functions will work during the test phase, will upload files before testing in case something happens. Couple of small1 errors got fixed, errors include removing head from getPrev call in the addStack function and forgot to add > in the addStack. Found an error where the stack didn’t go to the next node. Second test was a success.

**6/4/2019**

Worked on the queue header file today. Caught an error before compiling where I wrote getNext instead of setNext in my addQueue function. Just got done writing all of the functions needed for the queue class, didn’t take as long I originally thought it should, maybe because queues are very similar to stacks and linked lists when you look at it code wise. Going to commit code before testing.