Olliver Aikenhead

Lab Section: 1108

**Project 10**

The purpose of this project was to use templated classes with header files. The structure is to include just the header files without writing a .cpp file that goes with it. This project ended up being quite challenging because it was the first time I had worked with templates. Figuring out the template structure worked without getting compile errors and random code outputs was time consuming. I do have to admit, writing my own header file without the .cpp file was really cool. I didn’t think it was going to work at first.

I ended up getting some interesting compile errors while writing this project. One error was mainly figuring out the file structure without source files. Apparently on Mac, it is possible to compare and integer value with a size\_t value. This is not the case on Ubuntu 18.04 Bionic Beaver. I used a computer with Ubuntu on it I tested my code and saw all the compile errors from attempting to compare int with size\_t. It ended up being an easy fix to solve, I just had to change the compare values back to size\_t. After solving that problem, the program ran seamlessly on the Ubuntu machine. Another issue I had was getting the assignment operator to work. At first it was just printing junk, but after I changed the if statement to include the memory address. I always end up forgetting the memory address. I decided to utilize the push function for ArrayStack and NodeStack while writing the assignment operator. It ended up working out nicely. Another hard function to write was the copy constructor. I was not setting the m\_top variable to null when copying the data over during a function call. This was producing junk values as well, after setting the m\_top of the stack to null, it was not printing junk anymore.

To compile this project, make sure all of the files are grouped together when downloaded. Afterwards, go to the terminal and be in the correct directory and type “make all”. This will create an executable called “proj10”. Run the executable for the program output. This project was the most challenging by far since templates can be very confusing and written in many ways. I elected to not use template <typename T> for every function declaration. I found a way around this which I think makes the code look cleaner. I had to use the template <typename T> for the friend function between the classes.