Nick Alvarez

CS202.1101

Project 10

The purpose of this program was to take our linked list knowledge, both with literal nodes and also arrays, and implement the classes using templating so a DataType class was no longer necessary, being replaced by T. These classes utilize the FILO (first in last out) principle. Using array stacks and node stacks, it put new objects on top of the old ones only and removed objects from the top of the stack only.

Designing the program was nice as it retained simple push and pop functionality while simultaneously eliminating the need for DataType. Templating involved replacing anything that mentioned DataType as well as the forward declarations necessary for the insertion operator overload. Considering the entire class was written in the header file, it made writing the code take that much less time. Other than the templating and FILO structure, it functioned the same as the past few projects.

Conceptual issues with linked lists were still relatively scarce. Templates took some getting used to at first (ie. where to use them, how to declare them properly each time). Once that was taken care of, I implemented the rest of the project. One error, and I’m not sure how big of an effect it had, was the output of the program for the ArrayStack class. It would produce an extra zero at the end. Not one hundred percent sure about that one, and I experimented with the serialize function to no avail.

Given more time I would further investigate the output of the ArrayStack class. NodeStack seemed to work just fine, and the serialize functions are written in a similar way. There had to be something I’m missing but at least it still outputs and functions, so I cannot complain too much.