Logan Leavitt

CS 202.1103

Project 10 Documentation

This project aims to implement two varieties of the stack data structure. The first variety is array based, so the ArrayStack class is given a statically allocated array to hold its data. In contrast, the second variety is the NodeStack class and uses a linked list implementation that dynamically allocates its data through the Node class. In addition to this, the classes are designed as template classes, so they can hold any desired data type. Typing “make” will compile this project and typing “make clean” will remove the build files. The executable produced is name *proj10*. Upon running *proj10*, there will be four main repetitive sections of output. Each section is structured exactly the same, but tests a different set of stacks. The first two sections are dedicated to the Array based stacks, while the last two are for the Node based stacks. Each stack class is tested with both the int type and a char type to demonstrate the template functionality. To test a stack type I first test the constructors, which includes the the default constructor, parameterized constructor, and the copy constructor. I then print out the stacks to show that they were constructed successfully. The next few lines test the operator= method, top, and clear. I explain how I test those in the output. The few lines after that test the pushing and popping operations of the stack, which are the most essential to the stack data structure. I also explain how I test those in the output. I then test any miscellaneous functions such as size, empty, and full. This same structure is repeated for each major section of output to show that they work in a variety of cases.