Logan Leavitt

CS 202.1103

Project 9

To compile this project, you can type “make”, which produces an executable called *proj9*. To remove the executable and the object files, you can type “make clean”. The executable produced tests the ArrayQueue and NodeQueue classes required. Because both classes use all the same functionality, I put both of them through the same tests to show that all the required functionality works. The first section tests all of the constructors. The first list uses the default constructor, the second list the parameterized constructor, and the third list copies the second. This first section also shows that the operator<< overload and serialize method are functioning properly. The second section uses the front() and back() to assign values to the front and back of list three. I then test the operator= by assigning list three to list one. In the case of the Array-based Queue, it was really not necessary to implement the copy constructor, the assignment operator, or the destructor since this class does not work with any pointers or dynamic memory. The NodeQueue class, on the other hand, does need these functions implemented to prevent actual pointers instead of the underlying data being copied over, and to handle any dynamically allocated memory. The third section tests the functions which are essential to the Queue data structure. I push values and pop values in various ways onto the three different lists. For list one, I push two values and pop two values. For list two I push two values and for list three I just pop two values. The last section just tests the remaining functions: clear, full, empty, and size. I clear the contents of list one and then print the output of the other functions for each list. The Node Queue is run through the exact same tests as just described above. Additionally, I checked the executable for memory leaks and no memory leaks were found.