

Kripash Shrestha

CS 202 Spring 2017

Project 6 Documentation

4/17/2017

The purpose of this program is to test my ability to create and use Array Lists and Node Lists dynamic structures. I will need to be able to manipulate list-based dynamic data structures, classes, pointers, and iostream to all extents. My first attempt to work on this project was to look at the templates provided and project descriptions to create the needed functions.

I started with the Array List and had almost no problem with that. The only issue that I had with the Array List was with the find function. The original function modified the previous parameter locally so the variable was just changed or affected in main. I could not figure this out and had to go do professor PapaChristos' office hours to find out why it was not working. After I had figured that out I was easily able to work on my insert and remove functions for the Array List. For the empty function, I simply checked to see if m_size was equal to 0 or not. If m_size was equal to 0, the function would return true, if it wasn't then the function would return false. I did this because m_size would be updated constantly based on how much data is available on the array list. For the clear list, I just checked to see if the list was null or not; if the list was null, it does not need to be cleared, but if it was, then I would delete the list and set it to null.

I expected the Node List to be more difficult but it was a lot easier to deal with than the Array List for adding in and removing the element nodes. For the Array List it was a little difficult to

manipulate because of the size of the array and how the elements have to be relocated and properly shifted. However, for the Node List, all I did was find the target node and then hold the node's m_next member to a temporary variable. Then I created the new node to insert. And then I took the memory address of the new node and assigned it to the member m_next of the target node. Then I changed the m_next of the newly inserted node to the temporary variable that holds the m_next of the target node. That way the node list is continued and preserved through the connection of memory addresses. A very similar method can be applied to each function relating to insert and removing based on what you need to do. However, the Array List insert and remove functions had to account for each possible scenario that could occur with the desired outcome and present state.

If I could change something about my project if given more time, I would try to minimize my algorithm, steps and processes for certain algorithms and functions. I know I overdid something functions by overlapping some checks and doing some checks that would be confirmed by other functions, but I wanted to make sure and double check that everything worked. I also mixed array and pointer arithmetic for the Array List and I would want to focus on one or the other in the future. I prefer pointer arithmetic when it comes to manipulation of data structures but with experience I will be able to decide which and which is better for me and in general use.