Brandon Thibeaux

9/21/2020

CS-260-H1166

Reflection

The code that I have written takes a single linked list and allows a user to load data into the list, display data from the list (traverse), append, prepend, and remove data from the list. The main difficulty I had was really technical difficulties. Other than that it was really putting all of the material from the book to practice. Now that I’ve actually took the time to do it, challenge my self to go ahead of the video to see if I can make a solution without a guide, I’ve gotten a much better grasp on linked lists and pointers. I still have a long way to go before I get comfortable with them, but now they don’t seem so foreign. All of this is for making a software to keep track of auction information for this government entity. So, the reason why a linked list would be the best solution for this project would be because linked lists are dynamic and run much faster than arrays making them ideal for a situation where you could have thousands and thousands of records. Zybooks and the YouTube lecture really helped keep me on track when it came to producing code for this software. The only thing that I don’t think I would of done on my own would be initiating and housecleaning linked lists and some of the safety net code that was made in the YouTube video.