Dylan Dunagan

CS-300 DSA Analysis and Design

25 May 2025

Module 3 Assignment: Linked List Reflection

The purpose of this code is to create a linked list that reads from the CSV file and allows for the list to be added to, removed from, and searched through. The code is built with sections of classes and functions. The code allows for the creation of a new node, making it the tail of the linked list. It also allows for prepending which also adds a new node to the linked list but puts the new node as the head of list instead of the tail. Next, the code allows for printing the linked list, giving the user a visual output of what the list consists of. Next, the code allows for removing specific bids or nodes from the linked list regardless of where the nodes are placed. It then allows for a search function that will go through the linked list and find a specific node that the user is requesting. Finally, for the main method, it pulls all the functions previously specified together as a functional program that the user can then choose what the program does. Each case starts a different function, producing the specified output and giving the time it takes for the function to run its course.

As I worked through the participation problems in ZyBooks, I felt like I had a good grasp of how a linked list worked and what it does. I knew I needed to have an eye for detail as making sure the append, prepend, and remove functions are all written differently, but they all specifically state what the new “next” node is and what the new “previous” node is. That’s where I had most of my trouble in this assignment. It was making sure that I indeed had the correct logic in where nodes were being placed or moved to depending on what function was being called. To overcome this, I went back to ZyBooks and had the participation problems help me work through the logic step by step to ensure that the linked list was moving around appropriately.

The pseudocode for the main method is:  
Main function()

Loop that creates a menu for the user to choose cases as long as the user input is not “9”.

1 – Enter bid

2 – Load bids

3 – Display all bids

4 – Find bid

5 – Remove bid

9 – Exit

Case 1

Call the getBid() function

Add bid to the end of the list

Output the bid that was entered

Case 2

Call clock() function

Load the CSV file

Output the time it took for file to be loaded

Case 3

Calls PrintList() function

Outputs each linked list item

Case 4

Calls clock() function

Calls Search() function on bid specified by user

Loop that searches through each bid

If bid is found

Outputs specified bid

If bid is not found

Outputs that the bid was not found

Outputs the time it took for the list to be searched through

Case 5

Calls Remove() function on specified bid