CS-300

Code Reflection 3-2

Linked List Programming

Scott Dixon

I found the program this week to be similar to the Vector sorting code. The difference is that the linked list program is a data structure consisting of a sequence of nodes, each having a data element and a reference to the next node in the sequence.

This program consists of a LinkedList class, which contains a Node structure representing each node in the linked list. The LinkedList class provides methods for adding information and prepending nodes, removing nodes by ID, searching for nodes by ID, and printing the list.

One positive aspect of the linked list that I am finding is that the data structure has a dynamic nature. Linked lists can be resized when the program is run by adding or removing nodes, without needing to allocate a new block of memory. This makes linked lists useful for situations where the size of the data stored is not known. I also like that the program is modular. The LinkedList class provides a clear interface for interacting with the linked list, and encapsulates the implementation details of the linked list data structure.

However, there are also potential drawbacks to using a linked list. One negative aspect is that linked lists can be less efficient than arrays for certain operations, such as random access or iterating through the list in reverse order. This is because each node in the program has to be accessed sequentially.

Another negative aspect is the additional memory required to store the references to each node in the list. I am finding that the linked list program is a useful implementation of a dynamic data structure in C++, with its own set of advantages and disadvantages. While linked lists may not be the most efficient data structure in all situations, they can be a modular building block for various programs.

Pseudocode for linked list program.

START

Display program menu:

Enter a Bid

Load Bids

Display All Bids

Find Bid

Remove Bid

Exit

IF user selects "Enter a Bid"

THEN prompt user to enter bid details (bid ID, item title, fund name, and amount)

THEN add bid to linked list using Append method

IF user selects "Load Bids"

THEN prompt user to enter filename containing the text file is formatted for list of bids

THEN add bids to linked list using Append method

IF user selects "Display All Bids"

THEN PrintList is called to output all bids in linked list to console

IF user selects "Find Bid"

THEN prompt user to enter bid ID to search for

THEN call Search method to find bid with given ID

IF bid is found, display bid details

IF user selects "Remove Bid"

THEN prompt user to enter bid ID to remove

THEN call Remove method to remove bid with given ID from linked list

IF user selects "Exit"

THEN exit program with message

END