Name: Brandon Reid
Class Section: 1040.001
Lab Section: 307

UPDATE: Homework 4

- Make double link list
- Commands for left and right of current node

Homework Description

USE CLASS AND STL LIST - STL LIST IS A DOUBLY LINKLIST

Design a basic DOUBLY linklist structure using object oriented programming, with class structures.
The program will accept a list of commands, to modify the link list.

ADD
REMOVE
SEARCH
LEFT
RIGHT
PRINT
COMMANDS
EXIT

Program Steps:

1. Commands must be a string command input from the user
ex. Cmd> ADD X
    – I could make a string function to return the command to main
2. Start creating program by taking in the specific commands from the user and getting the name.
3. Switch statements for each commands would be ideal for clean program style
    a. since switch statements cannot take in string values, create an enum set of values for Strings
    b. then create a function to convert each command string taken in to a value
    c. you will have to make sure to take in the “name” the user enters before running conversion
4. Once user command input is calling to each case statement properly create a function for each command
5. ADD
    a. create a function prototype and definition within the class parameter to insert a Node

to the link list

b. make sure the insert function is inserting Nodes in alphabetical order, simple sort

c. make sure to have error syntax for duplicates and not add Node

6. REMOVE

a. create a function prototype and definition within the class parameter to remove Node in the link list

b. this will have to search for the Node before removing

c. this will also have to output an error message if name is not found for removal.

7. SEARCH

a. create function prototype and definition outside of class

b. search through list of nodes from front to back – while loop

c. if found output true, if not output false

8. LEFT

a. Create function prototype and definition outside of class

b. Go left one node of current node

9. RIGHT

a. Create function prototype and definition outside of class

b. Go left one node of current node

10. PRINT a. create function prototype and definition outside of class

b. iterate through list and print each node until end of list

c. print empty list if list is empty

11. COMMANDS

a. create function prototype and definition outside of class

b. cout list of commands for user

12. EXIT

a. create function prototype and definition outside of class

b. exit program with goodbye message