CPSC 121

Week 9
Singly-Linked-List

Lab 9 - Singly-Linked-List

Complete this lab by analyzing the given code. This code is a slightly modified excerpt from page 1042 of the textbook. Answer the following critical thinking exercises. Read through the entire program and compile it before answering the questions below.

1.) **What is this program doing?**

2.) **Take note of line 8. An overloaded constructor has been added to the struct. What is the benefit of having a constructor for a struct? At what line is this constructor being called?**

3.) **Analyze the loop in lines 25-30. Describe the steps that are happening.**

4.) **Analyze line #29, describe everything that is happening here.**

5.) **Analyze lines 33-46, what is this portion of the program doing?**

6.) **What is the role of *myCurrentNode, line #33?**

7.) **Analyze the loop in lines 39-44. Describe the steps that are happening.**

8.) **Analyze line #43, what is the purpose of this statement?**

9.) **Modify the program to add the sum of all the values and print the result.**

10.) **Write a function to check if the list is empty.**

11.) **Write a line of code that removes the first node of the list.** *(hint: how would you update the pointers?)*

```
#include <iostream>
using namespace std;

struct ListNode
{
   double value;
   ListNode *nextPtr;
   ListNode(double value1, ListNode *nextNode = nullptr)
   {
      value = value1;
      nextPtr = nextNode;
   }
};

int main(int argc, const char * argv[]) { //Note if you are on visual studio use int main()

   //BUILDING A LIST with a loop
   ListNode *head = nullptr;

   int number, input;
   cout << "How many numbers are in the list?" << endl;
   cin >> number;

   //BUILDS THE LIST from the head then makes the new node the head!
   for (int i = 0; i < number; i++)
   {
      cout << "Enter a num " << endl;
      cin >> input;
      head = new ListNode(input, head);
   }

 //TRAVERSE THROUGH A LIST with a loop to print the values
   ListNode *myCurrentNode = head; //Keeps track of current node to iterate through list

   int counter = 0;
   double sum1 = 0;
   cout << "Traverse the list of nodes!" << endl;

   while (myCurrentNode != nullptr)
   {
      counter++;
      cout << "Node " << counter << "'s Value is " << myCurrentNode->value << endl;
      myCurrentNode = myCurrentNode->nextPtr;
   }

   return 0;
}
```