Lab 2

Link Lists

In this lab, you will implement the Element module to create a working Double Ended link list.

**SPECIFICATIONS**

This workshop consists of the following modules:

* **Lab2.cpp** (Provided)
* **Data.h/Data.cpp** (Provided)
* **Element.h** (Provided)
* **Element.cpp**

The output from your executable running in Visual Studio should look like this:

**Elliott: (12345) - CompSci**

**John: (23456) - Engineering**

**Elliott: (12345) - CompSci**

**Jane: (23456) - Law**

**John: (23456) - Engineering**

**Mickey Mouse: (555) - Disney University**

**Elliott: (12345) - CompSci**

**Jane: (23456) - Law**

**John: (23456) - Engineering**

**Mickey Mouse: (555) - Disney University**

**Elliott: (12345) - CompSci**

**Danald Duck: (777) - Disney University**

**Jane: (23456) - Law**

**Mickey Mouse: (555) - Disney University**

**Elliott: (12345) - CompSci**

**Danald Duck: (777) - Disney University**

**Jane: (23456) - Law**

**Mickey Mouse: (555) - Disney University**

**Elliott: (12345) - CompSci**

**Danald Duck: (777) - Disney University**

**Mickey Mouse: (555) - Disney University**

**Danald Duck: (777) - Disney University**

Lab2.cpp

This file contains the main function (provided) and contains the code that will instantiation instances of your Element object and test the functionality of your double ended link list.

***NOTE: The main function SHOULD NOT be modified to make the output work. If the output is incorrect, the problem is in your Element.cpp file.***

Data Module (Data.h / Data.cpp)

The Data module (provided) is a Data class that stores the following information:

string Name (holds a students name),

unsigned int SN (student number), and

string Program (the program the student is enrolled in).

The public functionality provided to you is as follows:

Data() – constructor that will generate a blank (safe empty state) Data item that can be stored in an element of your link list

Data(string, unsigned int, string) – an overloaded constructor that will create a Data element with values for the students Name, student number and registered program.

void Display() – will display the contents of the data object

string getName() – returns the students name for use by your Element class to perform validation

Element Module (Element.h / Element.cpp)

The Element module is a class that creates a single item within your double-ended link list. The class is pre-defined for you in the Element.h file provided. You job is to create the Element.cpp file and implement the functionality required to make the Element class function correctly with the main. The following describes the functionality required:

Data Elements:

Data ElementData -- Data Object that contains the students name, number and program

Element\* pPrev – A pointer to the previous Element object in the link list. NOTE: If this is the head node it should be set to nullptr

Element\* pNext – A pointer to the next Element object in the link list. NOTE: If this is an end node it should be set to nullptr

Functionality:

Element(Data) – Constructor that takes a Data object and creates a new Element to be added to the link list. By default pPrev and pNext should be set to a safe state (nullptr)

Element\* PushFront(Element\*) – This function adds the provided Element object to the front of the link list and returns the address of the new “head” node

Element\* PopFront() – This function removes the first Element from the list and returns the address of the new “head” node

bool PushBack(Element\*) – This function iterates through the list to find the last Element. Then adds the provided element to the end of the list. If successful it returns true, otherwise false

Element\* PopBack() – This function iterates through the list to find the last Element. Then removes that element from the list and returns the address to the “head” node

bool addElement(string, Element \*) – This function iterates through the list to find the Element who’s name matches the string provided. It then adds the new Element to the right (i.e. pNext) and returns trues. If the Element who’s name matches the string cannot be found, it returns false

Element\* removeElement(string) – This function iterates through the list to find the Element who’s name matches the string provided. It removes that element from the list and return the address to it. If the Element who’s name matches the string provided cannot be found the function returns a nullptr

void DisplayList() – This function starts at the “head” node and iterates through the link list, printing out the contents of the Data within each element on a separate line. The data should be displayed on standard out (i.e. cout)

## Bonus Work

Update the Main function in Lab2.cpp to use a Standard Template Library (STL) List object rather than your Element class. This List should store the Data object just like your Element class did.