New Beginnings – Summer 2018

C++ Programming - Linked List Practice – Part II

This set of tasks builds on what you did in the first set. At this point, you should have two classes that look something like this(working code can be found in the course git repo at https://github.com/jasongraalum/NB_2018/Summer/code/LinkedList_Practice/PartI):

```
class List {
    public:
    Node * head;
    List();
    void addNode(int);
    void display();
};

class Node {
   public:
   int data;
   Node * next;
   void display();
};
```

Notice in the example class interfaces above that everything in each class is "public". This makes implementation easy since we can directly access the data members. However, often, as a developer of a class, we want to hide the data from the use and control the access to it. For part II, we'll move all the data members into the private section of the class and create a set of member functions to access the data.

Tasks

1. Move all data members of each class into the private section. (You'll have to create the private section for the classes above as it isn't included in the code I've given.)
2. Add member functions to each class so that you can do the following:
    a. Add a new node to the list. (The addNode member function in the Node class still works, but you'll now need one in the List class since you don't have direct access to the "head" data member.)
    b. Add the appropriate member functions to iterate through the list. This will require the ability to "get" and "set" a Node "next" pointer. (The "temp = temp->next;" line in the List::display() member function won't work since "next" is a private data member of the Node class.)
3. Update the List::display() member function to use the new access functions to the next pointer.