New Beginnings – Summer 2018

C++ Programming - Linked List Practice – Part III

This set of tasks builds on what you did in the first two sets of Link List Practice. 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/tree/master/Summer/code/LinkedList_Practice/PartII):

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

class Node {
  private:
    int data;
    Node * next;
  public:
    Node(int);
    Node();
    void setData(int);
    int getData();
    void setNext(Node *);
    Node * getNext();
    void display();
};
```

For the next set of tasks, we'll expand the Link List to hold additional data. Currently, each Node contains a single integer. In the Node of a Linked List, we can hold data in one of two ways. Either the Node itself contains the data(direct) or the Node contains pointers to objects with the data(references). The current implementation of our Node class is the former – containing the single integer.

For this part, we'll mix it up a bit and have the Node contain both direct data and references to data.

1. Create a new class called StudentRecord with the following **private** data members(all character arrays should be dynamically allocated):
    a. char * firstname;
    b. char * lastname;
    c. int credits; // gpa is credits divided by hours
    d. int hours;
    e. int dob_month;
    f. int dob_day;
    g. int dob_year;
2. Add a reference to StudentRecord to the Node class.
3. Rename the Node data member data to id. This is now the Students ID number. We will leave this as direct data in the node.

4. The Node class will now have three data members(all private):
    a. int id;
    b. StudentRecord * student_data;
    c. Node * next;
5. All the necessary member functions to the Node and StudentRecord class to set and access the all the data.
    a. Change the getData and setData function in the Node class to getID() and setID().
6. Modify the List function to insert Nodes sorted by the ID number(ascending.)
7. Add a display function to the StudentRecord class to output the data(well formatted.)