**Project 3**

**Lists and Object-oriented Programming**

This project is an individual assignment that focuses on object-oriented programming and lists. The list will be a doubly-linked list and it will be implemented using nodes and pointers. The data contained in the list will objects from a class.

* Project data: You will choose one of the following classes to define: Puppet, Herb, Marble, UnconventionalTheory, Hurricane, Store, Cat, Celebration, Employee, Airplane, State (*e.g.*, Nevada), EndangeredSpecies, and Shirt. Your class code will include a .h class interface file and a .cpp class definition file. You will choose the data attributes of your class. There should be at least 3 data attributes.
* Project data structure: Additionally, you will create classes for a doubly-linked list that will manage a collection of the objects from section (a). The list will use dynamically allocated nodes.
* For a user to work with your data and list, you should write a fully functional menu with the following options:
* Adding to the front of the list
* Adding to the back of the list
* Searching for and displaying a specific item in the list
* Editing a specified item from the list
* Removing an identified item from the list
* Displaying the entire list
* Ending the program

The following files are required:

The class driver similar to WorkingWithBooks.cpp (C).

The class declaration similar to Book.h (A).

The class implementation file similar to Book.cpp (A).

A header file for the node class similar to Node.h (B).

A header file for the list similar to SinglyLinkedList.h (B).

A class implementation file for the list class similar to SinglyLinkedList.cpp (B).

A word document that provides a list of the menu items (C.a through C.f) with the complexity of each operation using Big O notation.

Additionally follow all these project instructions:

* Your node file should contain your class name in it (*e.g.*, HouseNode.h)
* Your list file name should contain your class name in it (*e.g.*, HouseList.h and HouseList.cpp)
* Your program should run. Comment out any part of your file that does not work and briefly explain why you think it does not work. This documentation should also be added to the Word document.
* Submit (a) your Word document and (b) a .zip zipped project folder (not a .rar folder) which will include all the project’s files.

Grading [100 points total]:

Code should be documented and use meaningful identifiers 10 points

Code should be well-structured and easy to read 10 points

Files named properly and project folder zipped up properly 10 points

Adding to the doubly-linked list 15 points

Removing from the list 10 points

Searching the list 10 points

Editing the list 15 points

Overall running of the program 10 points

Complexity of menu operations 10 points