CS 202
COMPUTER SCIENCE II
SPRING 2016
Assignment #6

Due Date/Time: **04/26/2016 @ 11:59PM**
Total Points: **100**

Description

For this assignment you will extend the `LinkedList` class.

Specifications

Download the accompanying source code files and

1. Extend the `LinkedList` class by implementing the following functions:

➢
```
bool isSortedList() const;
//Function to determine whether the list is sorted.
//Postcondition: Returns true if the list is sorted in
//               ascending order, otherwise returns false.
```

➢
```
void sortList();
//Function to sort the list in ascending order.
//Postcondition: List is sorted in ascending order.
```

2. Complete the `TestLinkedList.cpp` file to test and display the functionality of the `LinkedList` class.
NOTE: You'll notice that `LinkedList.cpp` is included in `TestLinkedList.cpp` instead of `LinkedList.h`. This is because when templates are used, the compiler needs to generate the code where it is used in the program, so the class definition and implementation should both be included in `TestLinkedList.cpp`.

All programs must compile without errors and warnings on bobby.cs.unlv.edu using the g++ compiler. Programs that don't match these criteria will be given a zero (0).

No teamwork is allowed. All programs must be your own individual work.

Coding Style and Documentation

1. All submissions must have the following comment block at the top of their main program:

```
/*
 * Name: Your name, Class, Assignment number
 * Description: a brief description of the program.
 * Input: expected input to the program.
 * Output: expected output of the program. */
```

2. All functions and classes must have the following required documentation immediately above the function/class definition:

```
/*
 * function_identifier: brief description of what the function does.
 * parameters: what to pass into the function
 * return value: what the function returns, if any */

/*
 * class_identifier: brief description of the class
 * constructors: a list of constructor prototypes
 * public functions: a list of public function identifiers
 * private data members: a list of private data member identifiers
 * static variables: a list of any static variables */
```

3. All programs must employ proper indentation.

4. All programs must have reasonable comments throughout.

Submission

Submit your design document and source code files through WebCampus. You will submit the following four (4) files:

Design Document

- ✓ State the purpose and the functionality of your program.
- ✓ Include the specification of the class ADT for `LinkedList` class.
- ✓ Include a UML diagram for the `LinkedList` class.

LinkedList.h:

Specification file for the `LinkedList` class

LinkedList.cpp:

Implementation file for the `LinkedList` class

TestLinkedList.cpp

The file for program that tests and displays the functionality of the `LinkedList` class