**CMSC 3621**

**Lab 2**

Implement an additional method for “Linked list” as follows:

template <class List\_entry>

Error\_code List<List\_entry>::double\_insert(int position,

const List\_entry &x1, const List\_entry &x2)

/\*

Post: If the List is not full and 0 <= position <= n,

where n is the number of entries in the List, the

function succeeds:

Any entry formerly at position and all subsequent

entries have their position numbers increased by 2, and

x1 and x2 are inserted at position and position + 1,

respectively, in the List.

Else:

The function fails with a diagnostic error code.

\*/

Requirements:

1. Your implementation must be based on "Linked List" in the URL below:

<https://cs2.uco.edu/~gqian/cmsc3621/assignments>

1. Your implementation of double\_insert must handle pointers directly. You are NOT allowed to implement double\_insert by invoking insert twice in its body. A grade of 0 will be assigned otherwise. On the other hand, you are allowed to use set\_position in double\_insert.
2. The error codes provided by double\_insert should be similar to insert. For example, if position is out of range, range\_err should be returned.
3. Once you finish your implementation of double\_insert, you can uncomment lines 11-13 in main.cpp to test-run your implementation. The program should print the letters *a* through *h* in alphabetical order from the list if your implementation is correct.
4. Create a subdirectory named "lab2" under your server account on cs.uco.edu. Upload all your source code files and "makefile" to your server account under "lab2". The instructor should be able to use "make" to compile and test-run your program.

Submission:

The following files must be available under "lab2" in your server account on cs.uco.edu:

1. The source code files and "makefile" for "linked list" with your implementation of double\_insert.
2. The compiled, executable program "list"