

## CS260 - Programming Assignment 2 - Doubly Linked Lists and Templates

The purpose of this assignment is to give you some practice with implementing and using both doubly linked lists and templates

---

### Overview

For this assignment you'll write a program that uses two separate ordered doubly-linked lists implemented via a template (that you create).

You'll create a list of integers (from the template), and a list of instances of objects of your own design (from the template), and thoroughly demonstrate you can insert, remove, and print the items in the two lists.

When complete, you'll have the following files(your class names may be different):

- DoublyLinkedList.h (Template for creating doubly linked lists)
- A\_class\_of\_your\_own\_design.h & .cpp (you'll store instances of this in a list)
- DoublyLinkedListTest.cpp (routines to test your other code)

To do this assignment, I recommend doing the following::

- 1) Create a doubly-linked list that works with strings (similar to assignment 1)
- 2) Convert the code to a template. (remember, all template code goes in the .h file) and verify it works with ints and/or strings.
- 3) Create an ADT (class) of your own and make sure it works with the template as well.

That way you're not dealing with both doubly-linked list, template, and operator overload issues all at the same time..

---

### Details

The DoublyLinkedList template should be written so it can store a list of items of any type of object (that supports the necessary streaming operators). It should consist of the following methods (public functions):

- a constructor and destructor (if necessary)
- insert - inserts the given item in the list if it's not already there. Returns true if successful.
- remove - removes the given item from the list. Returns true if successful.
- count - returns a count of the number of items in the list.
- removeAll - removes all items from the list
- printInOrder - prints out list items, one per line, using streaming operator
- printInReverseOrder - prints out list items in reverse order, one per line, using the streaming operator

Main should declare two doubly linked lists, one using a simple data type (char, int, float, etc) and one using a datatype (class) of your own making. The class needs to consist of at least two private variables, whatever operators are necessary to be used with your DoublyLinkedList template, and whatever other functions are needed to make it work.

#### Additional details

- You don't need a menu (just call all the functions directly with test data)
- Your tests should be thorough, with the result being every line of template code is run.
- Unlike the example we did in the book, you make use of Node in the assignment, which contains a member of the data type the template is built around. So your node will need to be defined as follows:

```
template <class dataType>
class Node
{
public:
    dataType data;
    Node<dataType>* next;
    Node<dataType>* prev;
};
```

And everywhere you reference Node you'll need to append <dataType> to the end of it just as was done in the code above.

---

### Very Important Stuff

Program should be well written, function properly, and be both easy and efficient to use.

All programs should follow the class's [Coding Conventions](#)

Submit the following:

- A zip file containing all your .cpp and .h files and your executable
- The executable as a separate file (so the .exe gets uploaded twice)