

**CSCE 221 Cover Page**  
**Programming Assignment #2**  
**Due Date: Wednesday October 16, 11:59pm**

**Submit this cover page along with your report**

First Name: Bryant

Last Name: Passage

UIN: 326009948

**Any assignment turned in without a fully completed cover page will NOT be graded.**

Please list all below all sources (people, books, webpages, etc) consulted regarding this assignment:

CSCE 221 Students	Other People	Printed Material	Web Material (URL)	Other
1.	1.	1.	1.	1.
2.	2.	2.	2.	2.
3.	3.	3.	3.	3.
4.	4.	4.	4.	4.
5.	5.	5.	5.	5.

Recall that University Regulations, Section 42, define scholastic dishonesty to include acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion. Please consult the Aggie Honor System Office for additional information regarding academic misconduct – it is your responsibility to understand what constitutes academic misconduct and to ensure that you do not commit it.

I certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received nor given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment.

Today's Date: 10/14/19

Printed Name (in lieu of a signature): Bryant Passage

# Dequeues and Doubly Linked Lists

## Description:

For this programming assignment, you will implement a DEQUE whose size can grow as elements are inserted into the DEQUE. Specifically, you will implement a DEQUE with a doubly linked list. All of your functions should run in  $O(1)$ .

## Coding Portion (100 Points):

- Start with the template: Deque.h and fill in all of the member functions. Do NOT modify the definition of the functions of the DEQUE or put the declarations in a different file. We will be compiling the code you turn in with a set of tests and need your code to have the correct interface to compile the results. An example main.cpp that we will use to test the correctness of your data structure is provided.
- The implementation for the Node class has been provided.
- Be sure to test the correctness of your algorithms and implementations.
- When handling errors, please use an assert statement to indicate to the user what has gone wrong or throw an exception that indicates what has occurred. For example, if the user calls removeFirst(), assert that the Deque contains a node, or you can check if the Deque contains a node and throw an exception if it does not. The functions first, last, removeFirst, and removeLast should each throw exceptions under some circumstances.
- Your code will be graded based on whether or not it compiles, runs, produces the expected output, produces correct output, and your coding style (does the code follow proper indentation/style and comments).

## Grading Rubric

1. Valid file name as per submission instructions: 1 point
2. Executable code: 5 points
3. Doubly Linked List: 20 points
4. Each Function in Deque.h: 50 points (5 points each)
5. Correct output for all operations: 8 points (1 point each)
6. Organized and Readable code: 4 points
7. All operations in  $O(1)$ : 12 points (1.5 point for each)