COSC 2100 - **Assignment 4** (Fall 2019) Stacks and Queues **Due date: Oct 13, 2019**

A *double-ended queue* or *deque* is a generalization of a stack and a queue that supports adding and removing items from either the front or the back of the data structure.

This assignment has two parts:

**Part-1**

1. Create a doubly linked list based DeQueDLL class that implements the DequeInterface. The class skeleton and interface are provided to you.
2. Implement a String toString () method that creates and returns a string that correctly represents the current deque. Such a method could prove useful for testing and debugging the class and for testing and debugging applications that use the class. Assume each queued element already provides its own reasonable toString method.

**Part-2**

Create an application program that gives a user the following three options to choose from – insert, delete, and quit. If the user selects ‘insert’, the program should accept the **integer** input from the user and insert it into the deque in a sorted manner. If the user selects ‘delete’, the program should accept the **integer** input from the user and delete that integer from the deque. The program should continue to prompt the user for options until the user chooses to quit. Display the contents of deque after each insert/delete operation. You can make the following assumptions:

* User always tries to insert/delete an integer value only
* In case of repeated values, delete operation will delete the first occurrence only

Sample execution of code is provided on next page.

Ensure that your program works well by checking it against different test cases. At the minimum, submit three new and different test case reports with your program submission.

The **grading criteria** for this assignment are as follows:

|  |  |
| --- | --- |
| **Grading Criterion** | **Points** |
| **Does not compile** | **0** |
| Compiles | 8 |
| Input, Output and Computation is valid | 52 |
| Test cases | 20 |
| Coding Style | 10 |
| Documentation | 10 |

**Submission**: Combine all necessary files into a folder, zip/compress the folder and upload it on d2l.mu.edu dropbox before the due date. Also submit sample executions of different test case reports.

