**DATA STRUCTURE**

| | | |
|---|---|---|
| Type | ADT: Linked List | **LAB PERFORMENCE** |
| Deadline | **IN CLASS** | |
| Weighting | TBA | **4** |

**OBJECTIVES**

This assessment item is designed to test your skills on designing ADT such as Linked List, Linked Stack, and Linked Queue etc.

**ASSESSMENT TASK**

1. Implement the **Linked List** ADT which should include:
   (a) The definition of node and linked list class.
   (b) A default constructor which can initialize the linked list object.
   (c) A function `node* createNode()` which will return a new node created dynamically.
   (d) A function `node* getNodeAt(int position)` which will return the node at a specified position.
   (e) A function `void insertFront(int item)` which can insert an item to the front of the list.
   (f) A function `void insertBack(int item)` which can insert an item at the end of the list.
   (g) A function `bool empty()` which will return true if the list is empty and false if not.
   (h) An overloaded function `void insert(int position, int item)` which can insert an item at specified position. Before insertion it should check whether the specified position is a valid one. If not it should generate an error message as "invalid position".
   (i) A function `void deleteFront()` which can remove an item from the beginning of the list. Before deletion it should check whether the list is empty or not. If empty it should generate an error message as "list empty".
   (j) A function `void deleteBack()` which can remove an item from the end of the list. Before deletion it should check whether the list is empty or not. If empty it should generate an error message as "list empty".
   (k) An overloaded function `void delete(int position)` which can remove an item from a given position. Before deletion it should check whether the list is empty or not. If empty it should generate an error message as "list empty". Also it should check whether the specified position is a valid one. If not it should generate an error message as "invalid position".
   (l) A function `int getValueAt(int position)` which returns the item at specified position. Before doing so it should check whether the position is valid or not. If invalid it should generate an error message as "invalid position".

(m)A function **`void size()`** which returns the current size of the list.

2. Assignment:
   Implement the **Linked Stack** & **Linked Queue** ADT which includes all the previously defined functionalities of Stack and Queue respectively.

**WHAT & HOW TO SUBMIT**

You need to upload through your **VUES** account. You can find the upload link under "*Courses/ DATA STRUCTURE/Lab Performance/*"

**SUBMISSION STEPS:**

1. Create a Directory/Folder as following format:

   **<*Your ID*>_PERFORMENCE-< *Performence Number*>**

   Ex: 14-10380-1_PERFORMENCE-1
2. If you update your code then the format should be following:

   **<*Your ID*>_PERFORMENCE-< *Performence Number*>_UPDATE-<*Update Number*>**

   Ex: 14-10380-1_PERFORMENCE-1_UPDATE-1
3. Put all the files into that Folder and upload the **zipped** format of that Folder

**NOTES**

- Your submission will be rejected if uploaded in wrong format
- Only ".zip" file will be accepted.