DATA STRUCTURE

Type	ADT: Stack, Queue	**LAB PERFORMENCE**
Deadline	**IN CLASS**	
Weighting	TBA	**3**

OBJECTIVES

This assessment item is designed to test your skills on designing ADT such as stack and queue

ASSESSMENT TASK

1. Implement the Stack ADT which should include:
   (a) The definition of stack structure.
   (b) A function called **initialize**() which can initialize a stack object.
   (c) A function named **empty**() which will return true if the stack is empty and false if not.
   (d) A function called **full**() which will return true if the stack is full and false if not.
   (e) A function titled **push**(int item) which can insert an item on top of the stack. Before insertion it should check whether the stack is full or not. If full it should generate an error message as "stack overflow".
   (f) A function called **pop**() which can remove an item from the top of the stack. Before deletion it should check whether the stack is empty or not. If empty it should generate an error message as "stack empty".
   (g) A function named **top**() which returns the top item of the stack. Before doing so it should check whether the stack is empty or not. If empty it should generate an error message as "stack empty".
   (h) A function called **size**() which returns the current size of the stack.

2. Implement the Linear Queue ADT which should include:

   (a) The definition of queue structure.

   (b) A function called **initialize**() which can initialize a queue object.

   (c) A function named **empty**() which will return true if the queue is empty and false if not.

   (d) A function called **full**() which will return true if the queue is full and false if not.

   (e) A function titled **append**(int item) which can insert an item at the end of the queue. Before insertion it should check whether the queue is full or not. If full it should generate an error message as "queue full".

   (f) A function called **serve**() which can remove an item from the front of the queue. Before deletion it should check whether the queue is empty or not. If empty it should generate an error message as "queue empty".

   (g) A function named **front**() which returns the first item from the queue. Before doing so it should check whether the queue is empty or not. If empty it should generate an error message as "queue empty".

   (h) A function named **back**() which returns the last item from the queue. Before doing so it should check whether the queue is empty or not. If empty it should generate an error message as "queue empty".

   (i) A function called **size**() which returns the current size of the queue.

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.