6 DATA STRUCTURES

Lab Exercise # 6: Queue

Download the starter code from Brightspace and complete the following methods for a templated class called Queue. Please note that the **Queue** class should be an **array-based Circular Queue**:

1. `queue()` **(1 Point)**
   In the constructor, you may need to create the underlying array and initialize queue attributes. For example: front, rear, size, and capacity.

2. `~queue()` **(1 Point)**
   Inside the destructor, you need to delete the dynamically created internal array.

3. `void enqueue (T)` **(3 Points)**
   This method should add an element in the Queue. The internal array of the Queue should be used in a circular way for efficient utilization. This method should throw a custom exception called **QueueFull** when the queue is already full and an attempt to enqueue a new element is made.

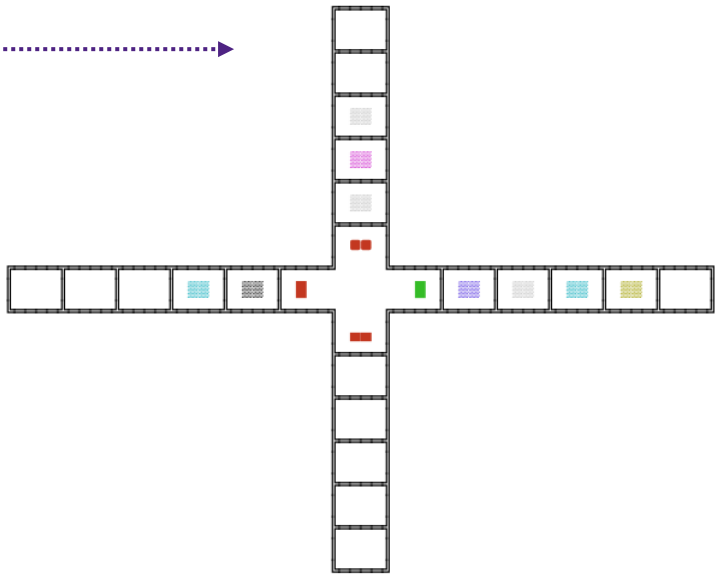
4. `T dequeue()` **(3 Points)**
   This method should remove and return an element from the front of the Queue. This method should throw a **QueueEmpty** exception when the Queue is already empty.

5. `bool isEmpty()` **(0.5 Points)**
   This method should return true if the queue is empty, or false otherwise.

6. `bool isFull()` **(0.5 Points)**
   This method should return true if the queue is full, or false otherwise.

Comments **(1 Point)**
Comments are a very important part of any program. You must always write comments in your code even if not explicitly asked.

Expected Output (simulation screenshot): ····▶



Note for Windows Users: If the program does not display correctly on your machine, you can run it using Ubuntu Virtual Machine on Oracle VirtualBox. A customized image of Ubuntu Virtual Machine can be downloaded from this Google Drive link.

You can get more help from Oracle Virtual Box Guide.

Alternate Option: You can also use Windows Subsystem for Linux (WSL) that allows to use Bash command line tools directly on Windows.

CODE OF CONDUCT

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment/lab-task.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your ideas with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi.

https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/