7 DATA STRUCTURES

Lab Exercise # 7: Vectors

In the lab exercise, you will complete a C++ program that enables users to perform various operations on a Vector. Download the starter code from Brightspace and complete the following missing methods for the template class **MyVector**.

Important!! Ensure that the vector resizes itself when the number of elements inserted into the vector increases beyond its capacity.

1. `MyVector(int cap=0)` **(1 Points)**
   Constructor creates a vector with the capacity passed as an argument. If no argument has been passed, then a vector of capacity 0 should be created.

2. `~MyVector()` **(0.25 Points)**
   Destructor performs the necessary cleanup (deleting dynamically allocated memory).

3. `reserve(int n)` **(1 Points)**
   If n is greath then the current capacity of the vector, the function should reallocate vectors storage increasing its capacity to n. In all other cases, no reallocation should occur, and the vector's capacity should remain unchanged. This function should not modify the vector's size or its elements.

4. `void push_back(T element)` **(1 Points)**
   This method adds an element at the end of the vector. If there is no space left in the vector then this method should resize the vector by calling the **`reserve`** method, before adding the new element.

5. `void insert(int index, T element)` **(1 Points)**
   This method inserts an element at the given index. It throws **`out_of_range`** exception if the index is larger than the `size-1`. The method should also resize the vector with the help of the **`reserve`** method when needed.

6. `void erase(int index)` **(1 Points)**
   This method removes an element from the index or throws **`out_of_range`** exception if the index is invalid.

7. `T& operator[](int index)` **(0.5 Points)**
   This method returns the reference of the element at a given index. The index can also be invalid.

8. `T& at(int index)` **(0.5 Points)**
   This method returns the reference of the element at a given index. The method should also throw an **`out_of_range`** exception if the index is invalid.

9. `const T& front()` **(0.5 Points)**
   This method returns a reference to the first element of the vector or throws a custom exception **`VectorEmpty`** if the vector is empty.

10. `const T& back()` **(0.5 Points)**
    This method returns a reference to the last element of the vector or throws a custom exception **`VectorEmpty`** if the vector is empty.

11. `int size() const` **(0.25 Points)**

This method returns the current size of the vector.

12. `int capacity() const` **(0.25 Points)**

This method returns the capacity of the vector.

13. `bool isEmpty() const` **(0.25 Points)**

This method returns true if the vector is empty, and false otherwise.

14. `void shrink_to_fit()` **(1 Points)**

This method reduces the vector's capacity to fit its current size.

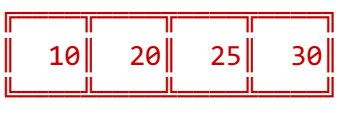
Comments: **(1 Points)**

Comments are a very important part of any program. You should always write comments in your code, even if not explicitly asked.

Desired Output:

```
> g++ lab7.cpp && ./a.out
List of available Commands:
display                  : Display the Vector
push_back <element>      : Add an element to the end of the vector
insert <index element>   : Insert an element at location index
erase <index>            : Remove the element from index
[index]                  : Returns a reference to an element at a specified index
at <index>               : Returns a reference to an element at a specified index
front                    : Return the (reference of) front element
back                     : Returns a reference to the last element of the vector
size                     : Returns the number of elements in the vector
capacity                 : Returns the capacity of vector
isEmpty                  : Tests if the vector container is empty
shrink                   : Reduce vector capacity to fit its size (shrink_to_fit)
exit/quit                : Exit the Program
>display
╔ size = 0
║
╚ capacity = 0
>front
Exception: Vector is Empty..
>back
Exception: Vector is Empty..
>push_back 10
┌────┐ size = 1
│  10│
└────┘ capacity = 1
>push_back 20
┌────┬────┐ size = 2
│  10│  20│
└────┴────┘ capacity = 2
>push_back 30
┌────┬────┬────┬────┐ size = 3
│  10│  20│  30│    │
└────┴────┴────┴────┘ capacity = 4
```

```
>insert 2 25
```




```
size = 4

capacity = 4
>insert 4 35
Exception: Vector index out of range...!
>insert 1 15
```


```
size = 5

capacity = 8
>front
10
>back
30
>at 0
10
>at 5
Exception: Vector Index out of range...!
>size
5
>capacity
8
>isEmpty
false
```

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/