Data Structures

Lab Exercise # 7 (Vectors)

(Points 1)

Download the **lab7_starter** code from Brightspace and complete the following missing methods for the Generic class ***MyVector***. Please make sure that the vector resizes itself if the number of elements inserted into the vector increases beyond its capacity.

- `MyVector();`
  No argument constructor that creates an empty vector with default capacity of 0
- `MyVector(int cap);`
  One Argument Constructor, which creates a vector with the capacity given as an argument.
- `~MyVector();`
  Destructor which does necessary cleanup of dynamically allocated memory.
- `void push_back(T element)`
  A method that adds an element at the end of the vector
- `void insert(int index, T element)`
  Insert an element at index, throws an exception if index is larger than then size-1
- `void erase(int index)`
  Removes an element from index, throws an *out_of_range* exception if index is out of the range.
- `T& at(int index);`
  A method that return reference to the element at index. The method should throw an *out_of_range* exception if index is out of the range.
  `const T& front();`
  A method that returns a reference to the first element
- `const T& back();`
  Returns a reference to the last element
- `int size() const;`
  A method that return current size of the vector
- `int capacity()const;`
  A method that returns the capacity of the vector
- `void shrink_to_fit();`
  A method that reduces vector's capacity to fit its size
- `bool empty()const;`
  A method that return true if the vector is empty, false otherwise

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 work 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.

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