

Data Structures

Lab Exercise # 8 (Iterators)

(1 Points)

The objective of this lab is to program a **custom iterator** for the DoublyLinkedList class written in lab exercise-4. The **iterator** will allow us to traverse our DoublyLinkedList in same manner as we can with STL List class (e.g. using range based loops).

You should add following additional methods into DoublyLinkedList class.

iterator begin()	Returns an iterator pointing to the first element in the list container.
iterator end()	Returns an iterator referring to the <i>past-the-end</i> element in the list container.

You should also add an inner-class/structure to the DoublyLinkedList class called **iterator**.

The **iterator** class/structure should contain following **iterator traits**:

value_type
pointer
reference
iterator_category
difference_type

The **iterator** class/should also should contain at least following **methods**:

iterator (Node<T>* node)	Constructor
T operator* ()	Operator to dereference the Node which iterator is currently pointing towards.
Iterator& operator++ ()	Prefix increment operator which makes the iterator point toward the next element in the container/list.
Iterator& operator-- ()	Prefix decrement operator which makes the iterator point toward previous element in the container.
bool operator==(const Iterator& other) const	operator that returns true if the current iterator is pointing towards the same element in the list which other iterator is pointing to.
bool operator!=(const Iterator& other) const	operator that returns true if the current iterator is not pointing to the element in the list which other iterator is point to.

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.

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