

EECS 330 Lab 2: Linked List

Objective

- Get familiar with doubly linked list basic operations and their implementations using C++.
- Get familiar with iterator.
- Get familiar with C++ inheritance and overloading.

Specification of the ADT

- Implement the basic linked list methods as summarized in the template file.

Hint: Implement the code available in Figure 3.11, 3.12, 3.13, 3.14, and 3.15 of the textbook (Data Structures and Algorithm Analysis in C++ by Mark Weiss, 4th Edition).

Additional Requirements

1. Rename the object name into “MyLinkedList”, instead of “List” as indicated in the textbook.
2. Implement `operator-- ()` and `operator-- (int)` for both of the `const_iterator` and `iterator` classes.
3. Implement a `void reverseList(void)` method for the MyLinkedList object, which reverses the current list (you do not need to keep the original list).
4. Implement a `MyLinkedList<DataType>& appendList(MyLinkedList<DataType>&& rlist)` method for the MyLinkedList class, which accepts as parameter another MyLinkedList class and appends all its items to the end of the current linked list.
5. Implement a `bool swapAdjElements(iterator& itr)` method, which swaps the element pointed by `itr` with its next element. If the swap is success, return TRUE; otherwise (e.g., the next element does not exist) return FALSE.

Testing and Grading

We will test your implementation using the tester main function posted online. The posted input and output examples should be used for a testing purpose, while we will use another set of inputs for grading. Your code will be compiled under Ubuntu 20.04 LTS using g++ version 9.3.0 (default) with C++11 standard.

Your final score will be determined by the success percentage of your program when fed with many random inputs. **Note that if your code does not compile (together with our tester main function), you will receive 0.** Therefore, it is very important that you ensure your implementation can be successfully compiled before submission.

Submission and Deadline

Please submit your implementation as a single .h file, with a file name “MyLinkedList_[YourKUID].h”. For example, if my KU ID is c123z456, my submission will be a single file named “MyLinkedList_c124z456.h”. Submissions that do not comply with the naming specification will not be graded. All submission will go through Blackboard. **The deadline is Friday Feb 24th, 2023, 11:59PM.**