

Lab 8 – CSCI 112

Due Date: Wednesday, Oct 21 at 11:59pm EST

Information

- This lab is intended to be completed **individually**.
- The files must be submitted with the exact file name provided in this file. If the file names do not match you will receive **zero** points for that file.
- Before you submit, make sure that your code runs. Any code which does not run without errors will receive **zero** points.
- Do not share your work with anyone other than Professor Khan or the TAs. You may discuss algorithms, approaches, ideas, but **NOT** exact code.
- If you submit work after a second past the due date **WILL** be locked out from submission.

Review

Lists

Lists are an ordered collection which allows random access to any of its contents. Lists allow the use of `[]` or subscript access to their contents. **ArrayLists** are most common, but **LinkedLists** can be implemented using a two way node.

List Iterators

List iterators are classes which are defined at the implementation level. These iterators allow for sequential mutation of a list, with most of its operations being **$O(1)$** . The iterator keeps track of where it is and where it was before, or if there is no valid previous location. The list iterators should be implemented as nested classes in your lists.

Nested Classes

Just like nested functions, nested classes are defined within another class. This both hides this class from outside access as well as ensures that the classes stay together. The list iterator classes are innately tied to the implementations, therefore should not be accessed by any other classes and should not leave their respective list implementations.

Task 1 – Testing Lists

[5 points]

Notice that this time you are not given the test cases for the lab and it will be part of your grade to create these files. Create a file named `testList.py` in your lab folder. Inside this file there should be two functions, `testList` and `testListIterator`. These two methods will take one parameter, the `listType`. This `listType` is the type of list implementation you wish to test out. As you program the following parts of the lab, add code to the appropriate function to test out the behavior and integrity of your code. Look to previous labs for examples of test functions. The points here will be awarded based on how robust your tests are (did you test all possible edge cases of all functions?).

To test for raised exceptions, use a try-except block (this should be in `testList.py`):

```
try:
    # Code that should crash, such as accessing an index out
    # of bounds or modifying inside an for-loop
except:
    # Print a message here saying it successfully crashed
```

Some example questions to help you inform your test cases: When programming insert, what happens if you insert at the start of the array? In the middle? At the end? At an index less than 0? At an index greater than the last index? etc.

Task 2 – LinkedList Pop()

[3 points]

LinkedList's pop method is not completed. Fill out the missing code in `linkedList.py`, making use of `__getNode` to obtain the node within `LinkedList`.

Task 3 – LinkedList Iterator

[7 points]

Open `arrayList.py` and examine how the nested `ListIterator` class is implemented (We also went over this in class). Create and implement a nested `ListIterator` class in the `linkedList.py` file in the same way `ListIterator` in `ArrayList` is implemented. Add testing conditions for these classes to `testListIterator` in `testList.py` as you complete the missing functions.

What To Turn In

Create a zip file named `Lab8_<your W&L ID>.zip`. Inside this zip archive should submit all the original files as well as the ones you created/modified.