

## CS151 Lab#3: Linked List

Reminder to have a TA check off on your lab exercises by the assignment 3 due date.

In this lab, we will learn about linked list and get ready for assignment 3.

**Exercise 1:** Design a class `City` that represents a city. It should have instance variables to store the following information. Include appropriate constructor, getters, setters and `toString`.

1. name of the city
2. population

**Exercise 2:** Implement a singly linked list that stores a list of `City`. Note, by convention, it should be called `CitySLL`. Code from class can be found in `~dxu/handouts/cs151/code/lec05`.

1. Create the appropriate `Node` class that supports `CitySLL`, as a nested inner class of `CitySLL`
2. The usual methods `size`, `isEmpty`, `first`, `last`, `addFirst`, `addLast`, `removeFirst`
3. Test the above methods to make sure they work properly
4. Override `toString` to print out a list of all stored city names
5. Add the following cities to the list: (Philadelphia, 1,567,442), (New York, 8,550,405), (Houston, 2,296,224), (Chicago, 2,720,546), (Los Angeles, 3, 971,883).
6. Print and make sure they are all there

**Exercise 3:** Implement a doubly linked list that stores a list of `City`. Note, by convention, it should be called `CityDLL`. Code from class can be found in `~dxu/handouts/cs151/code/lec06`. Your DLL should support the following functions:

1. Repeat steps 1-6 from Exercise 2 and make sure the doubly linked list is functional
2. Implement `insertBefore(City c, Node n)` so that you can insert a `City c` just before some `node n`
3. Use `insertBefore` to implement `insertSorted(City c)` so that a `City c` is inserted into the list in alphabetically sorted order.
4. Insert all cities use `insertSorted` instead. Print and make sure they are inserted in the correct order.
5. Modify `insertSorted` to insert in sorted order based on population instead