**CSE 274 – Fall 2018**

**PROJECT #6: 50 points – Due Sunday, October 28, by 11:59 p.m.**

**Outcomes:**

* Implement a List using a circular chain of linked nodes
* Refactor code written by someone else

## Naming requirements (not following any of these may result in a score of 0):

* The Eclipse project name must be **Project5**.
* You are given exactly four source code files:
  + **ListInterface.java:** an interface for the list ADT...do not modify this file
  + **LList.java:** a working implementation of ListInterface. This is provided for reference purposes. You do not need to change this file or turn in this file, but you may want to use it to see how things work.
  + **LListTail.java:** an exact copy of LList.java, except the name of the class has been changed. This is the file where you will do most of your work.
  + **Driver.java:** a tester that shows that LList.java is working correctly. You will modify this file to use your LListTail class instead.
* Use the **default package** (this means there should be no package statements in any of your files).

## Preliminaries:

* Review the list ADT (you should be familiar with list functionality from having worked with Java's ArrayList class).
* Try the Driver to see how the LList class works. Run the main() method in the Driver class to test LList.

## Assignment:

In a circular linked chain, the last node does not have a null reference. Instead, it refers to the first node in the chain. When you have a circular chain of nodes, it is not necessary to have references to the first and last node. Instead, all you need is to have a reference to the last node, because it is very easy to get to the first node from the last node.

Your assignment is to modify the code in LListTail.java as follow:

* Delete the firstNode instance variable, and all places in the class that use that variable.
* Create an instance variable: private Node lastNode. There should be only two instance variables in this class: Node lastNode, and int numberOfEntries.
* Modify the code so that the last node will always point to the first node (note that if there is only one node, then the first node is the last node...which means the last node would point to itself).
* Fix your code so that it uses this circular chain of linked nodes with a reference to the last node so that it correctly implements ListInterface.
* Modify Driver.java so that it tests LListTail instead of LList.
* Submit 3 files (all the files you were given, except LList.java).

**Scoring:**

|  |  |
| --- | --- |
| **Outcome** | **Max score** |
| LListTail implemented correctly | 50 |
| Code formatted according to generally accepted standards. | 0 (deductions only) |