CSCI 115
Lab Assignment 4
DUE: 3/1/15 (Sunday) at 11:59 P.M. on Mulan

For this lab assignment you will be developing a linked-list implementation with memory management. You will be implementing the ADT (abstract data type) that is provided in the List.h header file on Blackboard. Specifically you will need to:

- Create a Link template class for storing a single node of the list; this Link class should have
  - three **static** private member variables to keep track of the free list, the number of active nodes, and the number of free nodes. Because these are static, they are "global to the class": every instance of the class will share these variables (not have their own copies)
  - a constructor for the class
  - an overloaded new operator that grabs a Link from the free list if it is not empty and otherwise allocates a new one
  - an overloaded delete operator that puts the Link on the free list
  - the new and delete operators should be written in such a way that they correctly keep track of (in the static variables) the total number of active nodes and the total number of free nodes
- Create a LinkedList template class that inherits from the abstract List template class (in List.h) and uses Link to implement its virtual methods. These operations should have the following features:
  - The clear() operation should free (delete) all the Links in the list
  - prepend() and append() should allocate new nodes (which, of course, because of the overloaded new operator, should grab them from the free list, if possible)
  - numActive() and numFree() should return the values of the associated static variables of a Link object (any one of them, since they all share these variables)

These two class definitions will go into a lists.cpp file that #includes the abstract class header file (List.h).

In addition, you will need to create a driver.cpp file that also #includes the abstract class header file, and then runs a series of tests that exercises all of the features of your linked list implementation. The driver should print out information about which of your tests have passed and which have failed. **You will be graded based on how complete your testing is.**

Do NOT modify the List.h file!!!!