**CSCI 1101 – Winter 2013**
**Laboratory No. 9**

*This lab is a continuation of Linked Lists.*
*Your task is write and compile the class file followed by the tester program. Include all the source codes and outputs for each program. Please submit on Moodle by 12 noon on Saturday, March 23rd .*

Download the following files - Node.java and LinkedList.java (given next to the Lab9 link).
Write the following methods. These methods are all static methods and are to be used in a demo program (not to be added to the LinkedList class).

**Exercise 1:**

```
public static void downsize(LinkedList list, int n)
```

that removes every nth node from a linked list of Strings. Make sure to add the appropriate error checks.

Test your method in a demo program that creates a linked list and calls the above method for different values of n.

**Exercise 2:**

```
public static void reverse (LinkedList list)
```

that reverses a given linked list of Strings. Note that you should not create a new linked list but reverse the existing list.

**Exercise 3:**

```
public static LinkedList interleave (LinkedList list1, LinkedList
list2)
```

that takes in two linked lists of Strings and creates a third linked list that contains the items in the first two lists interleaved. For example, if
list1 → A → B → C → D → E → F and
list2 → W → X→ Y →Z then

list3 → A→ W→ B→ X→ C→ Y→ D→ Z → E→ F

that is, it adds the first item in list1, then the first item in list2, then the second item in list1, the second item in list2, etc. After it reaches the end of the shorter list, it just appends the remaining items from the longer list at the end.

**Exercise 4:**

```
public static LinkedList chopSkip (LinkedList list)
```

that removes every other node (keeping the first, third, ... nodes, removing the second, fourth, ...) from a list. The list
("aardvark", "bear", "cougar", "dog", "elephant", "fox")
should become ("aardvark", "cougar", "elephant")