CSCI 211 Lab 3. Queue

Lab Objectives

- To understand a Linked Queue implementation
- Be able to add an equals method to the provided implementation
- Be able to use a Stack object to implement a reverse queue operation

For this lab you will be provided with the LinkedQueue ADT and its implementation from the textbook (File/Import/Existing Project/Archive File/QueueExample.zip). You will add two methods to the LinkedQueue implementation: an equals method and a reverse method. The methods are briefly tested in the provided Driver class.

The method signatures have already been added to LinkedQueue and QueueADT:

```
public boolean equals(LinkedQueue<T> list);
public LinkedQueue<T> reverse();
```

Step 0: Study the Driver class. Note that for now, equals always returns true, and reverse returns null. Run the Driver program and note the incorrect output.

Step 1: Similar to any equals method, your equals method should return true only if the two lists have the same length, and the data elements of the nodes in "this" list and the list provided in the parameter list are also equal. Fill in the code for the equals method. *Be sure to use a temporary pointer (LinearNode<T>) to step through your list so you don't change the head pointer! Recall that LinearNode allows you to getElement (which in this case is a String).*

Step 2: For the reverse method, you should step through the LinkedQueue (again with a temp pointer) and push each element onto an initially empty ArrayStack. Then, while the stack is NOT empty, pop the elements off of the stack and enqueue them on a newly constructed LinkedQueue. This should result in a reversed list. The StackADT and ArrayStack classes have been provided.

Step 3:

1. Run your program for the lab TA
2. Export the project
3. Upload the zip file to Lab3 on Blackboard
4. Sign and turn in your lab sheet to the TA