Programming Assignment #2
CS 163 Data Structures

Submit your assignment to the D2L Dropbox (sign on via d2l.pdx.edu)

Goal: The purpose of the second programming assignment is to experience stacks, queues, and new data structures. The data structures for this assignment are a linear linked list of arrays and a circular linked list.

Problem Statement: A couple of weeks before I went to China, my mom had a stroke and we have spent the last month preparing for her change of care. Last week we completely moved her out of her independent living apartment into a POD. This week, we move her into assisted living, but the moving process is pretty intense. How we packed the POD was essential to the ultimate success of our project. Some of the contents will go to my mom's new home and others will go to friends or family. We had to be VERY careful how we loaded the POD. What went in first, needs to obviously come out last. The first in, ends up being the last out! We don't have time to shift through it all when unloading.

This fits many real world problems where UPS or other shipping companies need to be careful how they load their containers or trucks. The first delivery of the day should be easy to find – right at the doors of the container or truck. The last delivery, can be at the back of the container. This uses concepts of stacks and queues rather heavily.

Your job for this program is to write a simulation for packing PODs or containers for easy delivery. The information in YOUR containers will have information about the packages being delivered:

a. A sender (name, contact information)
b. The destination or recipient
c. The relative size of the package

We will then use the concepts of stacks and queues to simulate the packing the container and accessing the contents.

Programming – ADTs: There are two ADTs in this program (Queue and Stack ADT).

- Queue ADT will keep a list of recipients for delivery. The queue will help users like me figure out where to deliver the next item in the POD.

- The Stack ADT will keep the contents of the POD or container.

Please keep in mind that because we are creating ADTs, the **user** of the program must not be aware that stacks and queues are being used. Make sure to thoroughly test each of the stack and queue functions!

Programming – Data Structures:

- The stack should be implemented using a linear linked list of arrays that was discussed in class (and Lab 3). The array must be dynamically allocated. Each element of the array will be a package. Have the constructor of the stack class receive the size of the array. Do not use the subscript operator to access elements; instead use **pointer arithmetic** as we discussed in class. You must implement push, pop, peek, and display.
- The queue should be implemented using a circular linked list of recipients, where the rear pointer points to the rear, and rear->next points to the front. You must implement enqueue, dequeue, peek, and display.

Things you should know...as part of your program:

1) Lab #3 is on stacks, Lab #4 is on queues; they use these data structures.
2) **Do not use statically allocated arrays in your classes or structures.**
3) All data members in a class must be private
4) Never perform input operations from your class in CS163
5) Global variables are not allowed in CS163
5) Do not use the String class! (use arrays of characters instead!) You may use the cstring library.
6) Use modular design, separating the .h files from the .cpp files. Remember, .h files should contain the class header and any necessary prototypes. The .cpp files should contain function definitions. **Never "#include" .cpp files!**