**EE 205 Lab 9**

**Stack and Queue**

**Objectives**

The objectives of this lab are to learn how stacks and queues work.

.

**Task 1**

This task may be done in teams. Reverse Polish notation is explained in Lab 8.

Run [the reverse Polish calculator code.](https://dl.dropboxusercontent.com/u/1895560/UH-PUBLIC/EE%20205%20C%2B%2B/KnR_polish_calculator_example.docx) The code is at the end of the file. It is plain C code that implements HP calculator. It is using stack to implement it.

Write up your explanation as to how this code works in a short text file called RPN-as-stack.txt

When you are done, show it for TA for grading and feedback.

**Tasks 2 and 3**

Implement stack as array and then as list using the [Stack PPT](https://dl.dropboxusercontent.com/u/1895560/UH-PUBLIC/EE%20205%20C%2B%2B/PPTs/Stack.pptx) code (slides 6 and 7 at the moment) as the starting point.

Suggestion: when you implement stack as array, do not move elements around when you push or pop because that is too slow. Instead, move the indices marking the start and the end of the stack. That will make your stack “circular” i.e. it can wrap around the “ends” of the array, for example it can start at index 3, occupy indices 4, 5, … , N-1, 0 and end at index 1. Also, your stack is finite, i.e. bound by the size of the array, so you will not be able to push onto it beyond the maximum size.

Stack is always limited to some maximum size N. Lists can grow indefinitely. When you implement stack as a list, do not let your list grow beyond N elements. You may want to implement stack using a circular list.

Your stack should have push, pop and prettyprint. For right now, it is ok to have it as a stack of integers.

Call your files stack-array.cpp and stack-list.cpp. Use header files (because you are building stack library). Reuse your list library.

Write a driver main.cpp that will test your library.

When you are done, show it to TA for grading and feedback.

**Tasks 4 and 5**

Implement queue as array and then as list using the [Queue PPT](https://dl.dropboxusercontent.com/u/1895560/UH-PUBLIC/EE%20205%20C%2B%2B/PPTs/Queue.pptx) code as the starting point. There are already some codes provided for you there. Feel free to pick your “favorite version” and modify it to your liking. You will be using it for your next programming assignment.

The same specs as for the stack hold for the queue: make a circular array, and limit list size. The only difference between stack and queue is implementing pop/push versus enqueue/dequeue. In other words, copy over your stack code and modify it to be a queue.

Your queue should have enqueue, dequeue and prettyprint. For right now, it is ok to have it as a queue of integers.

Call your files queue-array.cpp and queue-list.cpp. Use header files, because you are creating queue library. Reuse your list library.

Write a driver main.cpp that will test your library.

When you are done, show it to TA for grading and feedback.

# Grading

* Put all your files into EE205/Labs/Lab8 directory.
* Copy and paste your codes into Laulima submit window or however your TA wants you to submit. Attach the tar file too.

Demonstrate to the TA that you have completed the code, for 4 programs you wrote. Total number of points is 50. You must finish the lab and show it to the TA ideally by the end of this lab and at most by the start of next lab.