# Assignment 8 (Templates)

A queue is a first-in-first-out data-storage technique (like a line at a bank teller's window). If one puts in 1, 2, and 3, one gets back 1, 2, and 3. (Remember that a stack is a last-in-first-out data-storage technique. If one puts in 1, 2, 3, one gets back 3, 2, 1.)

A stack needs only needs to keep track of one index for an array. A queue must keep track of two indices for an array. One index keeps track of the tail of the queue where new items are added. The other index keeps track of the head of the queue where old items are removed.
Let the initial size of the queue to be 10.
Develop and test the following methods:

- Queue operations:
  - enqueue(object): inserts an element at the end of the queue
  - object dequeue(): removes and returns the element at the front of the queue
  - object front(): returns the element at the front without removing it
  - integer size(): returns the number of elements stored
  - boolean isEmpty(): indicates whether no elements are stored
  - boolean isFull(): indicates whether queue is full

Write a class template for a queue class. Define several queues of different data types (char, int, double) and insert and remove data from them. Write a member function to print the queues. Use this function to verify the enqueues and dequeues.

# Assignment 8 (Templates)

Test your queue template with the following data:

Character queue:
Insert the following (enqueue(object)): a b c d e f
dequeue(): three times
Insert the following (enqueue(object)): g h i j
dequeue(): eight times

Integer queue:
Insert the following (enqueue(object)): 1 2 3 4 5 6
dequeue(): twice
Insert the following (enqueue(object)): 7 8 9
dequeue(): four times

Double queue:
Insert the following (enqueue(object)): 1.1 2.1 3.3 4.4 5.5 6.6
dequeue(): once
Insert the following (enqueue(object)): 7.7 8.8
dequeue(): five times

Print the queue after each operation (there should be about 25 output statements per data type).

Be sure to test the following:

1. The isEmpty() and IsFull() methods for both the pass and fail conditions
2. The front() method
3. The copy constructor

In addition to developing your own software (10 points possible), you can earn an additional 2 points if you satisfy the above requirements using the <queue> Standard Library class (STL).

# Assignment 8 (Templates)

If you used a linked list in the development of your queue, set a limit on the number of nodes in order to test if the queue is full.

Due March 25th