CS142 **Assignment 11** Spring 2025

Given: Thu, Apr 17
Due: Thu, Apr 23, 3:00 p.m.

1. (100%) A *queue* is essentially a waiting list. It's a sequence of elements with a front and a back. Elements can only be added to the back of the queue and they can only be removed from the front of the queue. Elements are kept in order so that the first element to enter the queue is the first one to leave it.

   Create a generic class of queues called `Queue`. Include the following operations. Make sure you read the instructions that follow the operations before you start coding.

   (a) A default constructor that initializes the queue to be empty.

   (b) A copy constructor. The elements of the constructed queue should be copies of the elements of the other queue.

   (c) An assignment operator. The elements of the queue on the left should be copies of the elements of the queue on the right.

   (d) A destructor.

   (e) A method `size()` that returns the number of elements in the queue.

   (f) A method `empty()` that returns **`true`** if the queue is empty.

   (g) A method `front()` that returns a reference to the front element of the queue.

(h) A method `pop()` that removes the front element of the queue. A copy of that element is also returned.

(i) A method `push(e)` that adds a copy of element `e` to the back of the queue.

(j) A method `clear()` that removes all the elements of the queue.

The easiest way to implement this class is to use an STL `list` to store the elements of the queue. But for the purposes of this assignment, *you are not allowed to do that*. Instead, you have to directly use a *singly-linked* list, that is, a linked list in which each node has only one pointer.

Note that all the `Queue` operations should run in constant time, with the exception of `clear`, the destructor, the copy constructor and the assignment operator.

Tip: When implementing most of these operations, draw pictures!

Split your class into header and implementation file in the usual way. Submit your test driver. Make sure to test every operation.