COEN 79L – Object-Oriented Programming and Advanced Data Structures

Lab 8.5:

You will implement and test a **PriorityQueue** class, where the items of the priority queue are stored on a linked list.

The standard queue retrieves element in the first-in first-out ("FIFO") manner. However, this strategy is not always what is needed. A hospital emergency room, for example, needs to serve patients according to their priority. A patient with a more critical problem will pre-empt others even if they have been waiting longer. This is a priority queue, where elements are prioritized relative to each other and when asked to dequeue one, it is the highest priority element in the queue that is removed. There are many applications for priority queue.

The Node class for this linked list must be defined in your header file for the priority queue rather than in a separate header file. All of the PriorityQueue member functions should be written by you from scratch (not using the linked list toolkit).
Note: This is not a template class; you will use a `typedef` statement to define the underlying data type.

Purposes:
Give you more practice in building and manipulating linked lists.
Introduce you to priority queues.

Files that you must write:

1. `pqueue1.h`: The header file for this first version of the PriorityQueue class. Actually, you don't have to write much of this file. You just need to finish the PriorityQueue class definition. If some of your member functions are implemented as inline functions, then you may put those implementations in this file too.
2. `pqueue1.cxx`: The implementation file for the PriorityQueue class. You will write all of this file, which will have the implementations of all the PriorityQueue's member functions. Also, remember that the PriorityQueue's linked list consists of dynamic memory, so you will need to define a copy constructor, an assignment operator, and a destructor.

Other files that you may find helpful:

1. `pqtest.cxx`: A simple interactive test program.
2. `pqexam1.cxx`: A non-interactive test program that will be used to grade the correctness of your PriorityQueue class.