Assignment 1: Priority Queue

Due: Friday January 19th, 23:59. Worth 3% of your course grade.

Web stuff: See the course web site for starter code, and a summary of any hints and announcements about the assignment.

Purpose: Review C++ basics, including a class that contains instances of another class; operator overloading; friend functions. Introduce the use of an abstract class to represent the interface for an ADT, and a child class to represent the implementation of the ADT.

Priority Queue: For this assignment, you will implement a priority queue. A priority queue is a list of items that can be compared according to their “priority”. We can do the following things to a priority queue:

- **enqueue(item):** Put item into the priority queue.
- **dequeue():** Return an item from the priority queue, and also remove it from the queue. The item returned should be the one with the highest priority. If there is a tie for top priority, the item that entered the queue first (among those that are tied) should be chosen
- **size():** Return the number of items currently in the priority queue.

“Priority” can be anything according to which the items can be ordered. For example, you could have a priority queue of customer purchases, and treat price as the priority, with highest price being top priority. When you remove something from this priority queue, the purchase with the highest price would come out first. (Or if several purchases have the same highest price, the oldest of them would come out first.)

To implement the priority queue ADT properly in C++, one should use templates — so that the priority queue can contain any sort of items that can be prioritized. For this assignment, however, we will avoid templates by dealing only with priority queues that contain to-do items. A to-do item is simply a string, like “clean the bathroom”, and an integer priority, like 12. We will treat 1 as the highest priority.

Starter code: You will be given the following starter code on the course web page:

- **PriorityQueue.h:** An abstract class that defines the “interface” to a priority queue.
- **LinkedPQ.h:** A child of class PriorityQueue that “implements” the interface set out by PriorityQueue.
- **TodoItem.h** and **TodoItem.cc:** These define a very simple class for to-do items.

**You must not change any of the code I provide.** If you do, your program may fail all of the autotesting.

Your task: Your task is to implement the priority queue ADT using a linked list. That is, write the bodies of the functions declared in LinkedPQ.h, putting your code in a file called LinkedPQ.cc. You will need to define your own class for nodes in the linked list.

How to submit: Details on how to submit your assignment, as well as how to “exercise” it will be available on the web site.