Queue

Date assigned:	Monday, October 17, 2011
Date due:	Friday, October 21, 2011, 11:59 pm
Points:	40

For this assignment, you are to implement the Queue ADT in a file called Queue.c using the header file Queue.h. You can find this header file on Zeus in **/home/CS300Public/2011**. All of the data structures and function prototypes are defined in the header file. Further, each function prototype has been described to the point that you should be able to implement each function. The error codes that can be produced are listed for each function. Higher precedence error codes are listed first.

The Queue must be implemented using the Priority Queue from assignment 5 as the base data structure. Your Queue should contain a PriorityQueue in the same manner that a PriorityQueue contains a List. We will be reusing this Queue later on, so make sure you have completely tested and debugged each operation. No new pQueue.h or list.h is necessary for this assignment.

In addition to implementing the Queue data structure, you must provide a Makefile and test driver (queueDriver.c that produces an executable named queueDriver) that thoroughly tests your Queue. The queueDriver should be similar in thoroughness to the stack driver we discussed in class. The driver should display to the screen a series of SUCCESS or FAILURE messages with enough description that a user can quickly spot broken functionality.

You may add any helper functions you need to Queue.c. These helper functions should be marked **`static`** so they are not available outside the module. You may not alter Queue.h in any way, **except** to change the ***`#include "… pQueue.h"`*** line.

1. Your code is to be written in C using Eclipse 3.7. Programs written in other environments will not be graded. Create an Eclipse project named cs300_staticQueue_PUNetID. This project should contain the directories: src, include, and bin. The driver, QueueDriver, should be created at the top level of the project, not in the bin directory.

2. The Makefile must contain the necessary targets to build the queueDriver as well as a clean and dist target similar to the identically named targets in your Stack assignment. There is no testMe target required for this assignment. Typing **`make`** on the command line should build queueDriver.

3. Submit a file called cs300_staticQueue_PUNetID.tar.gz into the CS300 Drop Box by 11:59pm on the day in which the assignment is due. This file must include your Queue, your Priority Queue, and your Static List project.

4. Submit a color, double sided, stapled packet of code by that same deadline. The packet should be in the following order:
   Makefile
   Driver (.h then .c if you have both, otherwise just .c)
   Queue.c (do not print Queue.h)

Any extra .h/.c pairs you have. (do not include any code from the Priority Queue or list project)

5. Test one function at a time. This will lessen your level of frustration greatly.
6. You are to use the coding guidelines from V6.0 of the coding standards.
7. The only changes to StaticList and pQueue you can make are to fix bugs.
8. You must insert your Queue project into your Subversion repository.

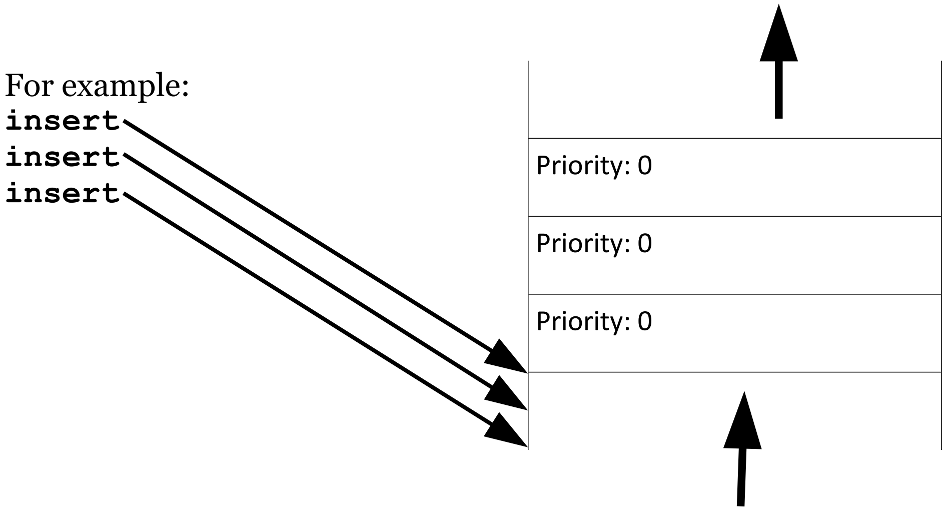
Goals for this assignment:

1. Reuse your Static List and Priority Queue.
2. Code and test your program one function at a time.
3. Write efficient/clean code
4. Use the debugger to effectively develop a correct solution
5. Thoroughly test your code.

Masking Priority in the non-Priority Queue.

Your Queue **queueEnqueue** function accepts from the user a **`Q_DATATYPE`**. The user does not specify a priority to **queueEnqueue.** The **queueEnqueue** function must then call **pqueueEnqueue** to add an element to the underlying Priority Queue. The **queueEnqueue** function must always provide a priority of zero to **pqueueEnqueue**. This will cause your Priority Queue to default to non-priority queue behavior.

For example:



There is only one deadline. I expect you to start this project soon. Again, your queue data structure will likely be smaller than your queueDriver.

Your StaticList driver and PriorityQueue driver must still compile and run.