

Queue

Date assigned: Wednesday, October 24, 2012
Date due: Wednesday, October 31, 2012
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/2012**. 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. No new pQueue.h or list.h is necessary for this assignment. We will be reusing this Queue later on, so make sure you have completely tested and debugged each operation.

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 list 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. Programs written in other environments will not be graded. Create an Eclipse project named cs300_queue_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. Typing **make** on the command line should build queueDriver.
3. Submit a file called cs300_queue_PUNetID.tar.gz into the CS300 Drop Box by 9:15am on the day in which the assignment is due. This file must include your Queue, your Priority Queue, and your List projects.
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)

- A list of bugs fixed in pQueue and List: A short 2-4 sentence description of each bug which must include the name of each function updated for each bug.
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 List 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 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.

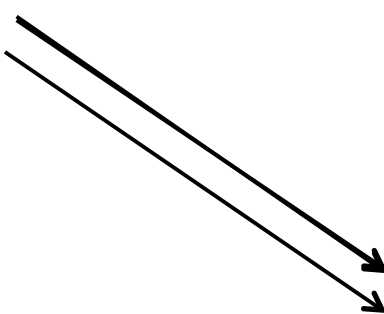
The queue.h header file is attached.

Masking Priority in the non-Priority Queue.

Your Queue enqueue function must call **pqueueEnqueue** and always provide a priority of zero. This will cause your Priority Queue to default to non-priority queue behavior.

For example:

insert
insert



Priority: 0	
Priority: 0	
Priority: 0	

insert

◆ **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 List driver and PriorityQueue driver must still compile and run.