**CSI 403 – Spring 2018 - Project #2 – Priority Queue**

Goal:

* To simulate a task management system with a priority queue.

Problem:

* Provide a RESTful service which accepts as a POST of JSON a list of enqueue and dequeue statements onto an in-memory job queue.[[1]](#footnote-1)
* Each job definition contains a name and a priority, with 0 being the best priority and positive integers representing lower priorities.
* Return the JSON representing the state of the queue (the list of job names, in priority order), after all enqueue and dequeue statements have been processed.
* Example input: { “inList” : [ { “cmd” : “enqueue”, “name” : ”job1”, “pri” : 4 },

{ “cmd” : “enqueue”, “name” : ”job2”, “pri” : 3 },

{ “cmd” : “dequeue” },

{ “cmd” : “enqueue”, “name” : ”job3”, “pri” : 0 },

{ “cmd” : “enqueue”, “name” : ”job4”, “pri” : 1 },

{ “cmd” : “dequeue” }

] }

Example output: { “outList” : [ “job4”, “job1” ] }

* Erroneous input (e.g. malformed JSON) should be handled gracefully.

Deliverables:

Submit to the Blackboard by the due date:

* An HTTP URL to a RESTful service which must remain up and running 24/7 until grading is complete. Graders will invoke your service with a tool such as curl or Postman at a time of their choosing.
* A ZIP file containing your source code, written in any language you choose.

//natural ordering EXAMPLE of priority queue

* Queue<Integer> **intPQ** = **new** PriorityQueue<>(7); //size 7
* **Random** **rand** = **new** Random();
* **for**(**int** **index** = 0; index < 7; index++)
* {
* intPQ.add(**new** Integer(rand.nextInt(100)));
* }
* **for**(**int** **index** = 0; index < 7; index++)
* {
* **Integer** **in** = intPQ.poll(); //remove and return value
* **System**.***out***.println("Processing Integer:" + in);

}

1. No persistent state is implied – the RESTful service is stateless. [↑](#footnote-ref-1)