COSC 311 Programming Project #1 Multiple Server, Single Queue

Distributed: 1/29/2015 **Due**: 2/18/2016

Precis: Simulate a multiple server, single queue system. Collect average wait time and total wait time for different number of servers. Considering cost of server versus increased service, make a recommendation as to number of servers.

Problem description:

You own a consulting company: *Q & C Solutions*. Your customer, *Yummy Donuts*, wishes to know whether it is worth the expense of acquiring one, two, or three new servers (counter clerks). Here are the particulars:

- One server has an ongoing cost of $100 per day.
- Each client that is served provides $1 profit that can go toward the cost of the new server(s).
- A typical 8 hour day will have 100 clients. The clients arrive at the queue in a Poisson distribution. As an approximation to Poisson, you can use the algorithm given by the text, section 4.5, to model the number of arrivals per tick.
- The service time per client is a random number of minutes between 1 and 5 (Gaussian distribution).
- A client that is in the queue for 5 minutes will drop out of the queue before being serviced. That client's wait time is *not* included in the average and total wait times. (Note, you should not actually remove the timed-out client from the queue. When a timed-out client reaches the head of the queue, *remove it without servicing it and do not include its wait time* (5 minutes) to the statistics.

Experiments:

Run the simulation 6 times, that is, 2 times for each number of servers k = 1, 2, 3. Each simulation should be for 8 hours.

Collect the total wait time, average wait time, number of clients served, number of clients that dropped out of the queue. Fill in the following table.

Note – this table can be filled out by hand, using the final results of each simulation.

Simulation	**k (# servers)**	**# Served clients**	**Total wait**	**Average wait**	**# Dropped clients**
1	1				
2	1				
3	2				
4	2				
5	3				
6	3				

Quality control:
Do an abbreviated simulation, one for each k = 1, 2, 3 (i.e., simulate once for one server, once for two servers, once for three servers) for 16 minutes. Provide verbose output, something like that in Figure 4.17. The verbose output must include:

- minute by minute trace of :
  - server activity ("idle", "serving" – give client #)
  - a list newly arrived clients with client # and service time
  - new queue size (include timed out clients)
- total number of clients entering queue
- total number of clients serviced
- total number of clients that timed-out
- final size of queue (includes timed-out clients)
- total wait time (not including timed-out clients)
- average wait time

Report to the customer:
Write one paragraph (100 – 300 words) summarizing the experimental results for 8 hour simulation; you need to make use of the numbers in the table that you filled out. Do a quantitative assessment, comparing cost/benefit for k = 1, 2, 3 servers. Make a recommendation to the customer regarding how many servers he should acquire.

Turn in:

- Hard copy of source code
- UML
- Hard copy of quality control runs
- Report to customer, including:
  - Table of results
  - One paragraph as described above in "Report to customer"

Note:

- You must use a queue data structure. You may use the Java Queue interface.
- You may, if you wish, implement your own Queue as LinkedList or ArrayList or Array.
- Your code must include javadoc comments; do not turn in javadoc pages.
- Your code must be readable and elegant. Good use of white space, good choice of names, no unnecessary computation.
- Be prepared to give demo and code walk through.

Grade based on:

- Meeting specs
- Quality of code (elegant, readable, alignment, documentation)
- Quality of Report to Customer (correct, readable, no English errors)