

# CS121 - Computer Science II

## Lab Assignment #6

### Fall 2011

The purpose of this exercise is to implement a dynamic queue, as well as to do some simulation programming.

The owner of the local fast lube has asked you to simulate his operation. He is considering adding more service bays, and wants to know how this will help the efficiency of his operation.

Cars arrive at the store, and enter a single queue, where they wait for service. The following facts are known:

- There are three service bays for changing oil.
- An oil change takes 15 minutes.
- A car arrives randomly between every 1 and 20 minutes.
- There is only one queue - the single queue feeds all three bays.
- The shop is open from 9:00am to 6:00pm every day.
- At closing time (6:00pm), no more cars are accepted into the queue, but those in line at that time are serviced until the queue is empty.

You are to write a program, using a dynamic queue, that simulates the operation of the store. You should determine the average waiting time (in minutes) in the queue for customers, and the amount of time any of the bays are idle each day. Do the simulation for a total of 12 months, and report on the results for each month. You can assume that each month is 30 days long, and a year consists of 12 months (360 days).

#### Some Hints:

- Pseudo-random numbers are often used in simulations like this. There are several random number generator functions available - one such is `random()`. The numbers are not really random - the functions actually calculate the next "random" number from the previous one. They are designed such that they have very long *periods*, meaning that the numbers they generate don't repeat for a very long time - this is also not a random characteristic, since if the values were truly random there would be a (very small) probability that the same number could come up twice. However, since the next value is actually calculated from the previous value, a sequence with a long period prevents the function from calculating the same set of pseudo-random numbers, at least for a very long time.

Since the new random number is generated from the old one, an initial *seed* value is used the first time the random function is called. This seed value is the same each time the program is executed, and therefore the same "random" sequence will be generated

each time, unless the seed value is changed for each execution. The seed can be set with `srandom(seed)`; Often, the value used for `seed` is some representation of the time of day, which, while not random, will at least change for each execution. An integer that represents the time of day (actually, the number of seconds past Jan 1, 1970 on UNIX/Linux systems), can be obtained by calling the function `time(NULL)`.

Finally, the `random()` function returns a value between 0 and `INT_MAX`. This value usually must be scaled according to the particular program needs. In our case, we need to scale the random number so that it produces a value between 1 and 20, to simulate the time when a new car will arrive.

- The queue of cars must be implemented dynamically. The important fact about each car is its arrival time - that time can then be compared to the time when it enters a service bay to determine how long it had to wait for service.
- The time resolution of your simulation can be one minute. For each minute, you will need to update the time, determine if any bays become empty, move any waiting cars into the empty bays, and determine if any new cars arrive. It might be useful to note that the shop is open 9 hours or 540 minutes each day.

An example output might look like:

#### Fast Lube Simulator

```
Month 1 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 2 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 3 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 4 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 5 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 6 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 7 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 8 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 9 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 10 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 11 - Average wait time: xxx minutes, average daily idle time: xxx minutes
Month 12 - Average wait time: xxx minutes, average daily idle time: xxx minutes
```