NAME:_____________ Student ID:_________

Department of Computer Science

Computer Science 177: Modelling and Simulation

FINAL EXAM — Winter 2013

1. Indicate whether each of the following statements is true or false. (2 points for each correct answer, -1 point for each wrong answer, 0 points for each blank.)

_____ a. If a $Q-Q$ Plot is used to compare the empirical and theoretical cumulative distribution functions for customer waiting times, say, then the units on both axes must be probabilities ranging from 0 to 1.

_____ b. In the *K-S Test* for ‘‘goodness of fit’’, the *maximum vertical distance* between the theoretical and empirical cumulative distribution functions *always* occurs at one of the measured data points.

_____ c. It is dangerous to choose the *normal distribution* for representing the amount of time required to complete some task because it may produce both negative and positive values.

_____ d. Since the Chi-Squared Test lets you decide how to split the data into non-overlapping intervals, there is no single ‘‘right answer’’ for the *goodness of fit* between one set of empirical data and a specific theoretical distribution.

_____ e. According to the principle of maximum entropy, the most convincing results are generated by *very large simulation models* that are so complicated that nobody can understand them well enough to determine their validity.

_____ f. *Synthetic variables* bring information about the global environment *into* a simulation model, whereas *antithetic variables* export information from the simulation model back to the global environment.

_____ g. A *CSIM-19 mailbox* might have a queue of sent messages, or a queue of processes attempting to receive a message, but never both at the same time.

_____ h. If the *Regenerative Method* is used to split a very long simulation run into multiple pieces, then the measurements from different pieces *are mutually independent*, but *not identically distributed.*

_____ i. If $X_1, X_2, \cdots$ is a sequence of i.i.d. samples from an arbitrary *continuous distribution* with PDF $F_X(x)$, then $U_1 = F_X(X_1), U_2 = F_X(X_2), \cdots$ will be an i.i.d. sequence of $U(0,1)$ random variables.

_____ j. A *Linear Congruential Random Number Generator* turns an initial ‘‘seed’’ into a sequence of integers between 0 and $m-1$ that *never* repeats (similar to the digits of an irrational number, like π).

2. Suppose you ran your elevator simulation program N times with different random number seeds to generate N independent measurements of the average travel time

$$X_1,\ X_2, \cdots, X_N$$

a. State the correct formula for the 95% confidence interval for the *global average* travel time across all N runs assuming that each run of your simulation program was long enough to assume that the average travel times across different runs have an i.i.d. normal distribution. [HINT: Remember to use the t-distribution, because your formula only has access to the sample variance.]

b. Suppose you increase the total number of runs, N, from 9 to 36. How does that affect the confidence interval in part (a), assuming the sample mean and sample variance do not change significantly.

c. Now suppose you must reduce the width of the confidence interval to 1/10th of its size when $N = 36$. Estimate *how many additional runs* of your simulation will be needed to satisfy this requirement.

3. The following segment of CSIM code could be used to represent actions of a single car in the gas station problem discussed in class.

```
1         void car()
2         {
3             create("car");
4             double litres_req = uniform(10.0, 60.0);
5             long queue_I_face = gas_station.qlength();
6             if ((queue_I_face == 0) ||
7                  bernoulli((40 + litres_req) / (25 * (3 + queue_I_face))))
8              {
9                 gas_station.reserve();
10                hold (normal(150 + 0.5 * litres_req, 30));
11                litres_served = litres_served + litres_req;
12                gas_station.release();
13                cars_s.note_passage();
14             }
15            else
16             {
17                litres_missed = litres_missed + litres_req;
18                cars_b.note_passage();
19             }
20        }
```

a. What would be the effect of deleting the `create()` statement at line 13?

b. Suppose every customer who decides to leave without purchasing any gas comes back later in the day for *exactly one more attempt* before giving up completely on buying gas from this station. Assume that the elapsed time between the customer's first and second attempt is uniformly distributed between 1 and 2 hours, and that the amount of gas required increases by exactly 5 litres at the second attempt. Modify the above CSIM code to include the second attempt.
[HINT: Don't forget that the program uses *seconds* as its time unit.]

c. Forty years ago, most gas stations provided *full service*, where a single employee (known as a ‘‘gas jockey’’) operated the gas pumps, and cleaned the windshield and checked the engine oil for each customer as he waited for their pump to finish. In other words, once a customer arrives at a gas pump, he simply waits until the ‘‘gas jockey’’ is ready to serve him. Briefly explain the relationship between the structure of a ‘‘full service’’ gas station with one ‘‘gas jockey’’ and a building with one elevator. Describe the changes you would make to adapt the structure of your elevator model to this new situation. Do *not* attempt to write the entire code for the ‘‘gas jockey’’ process — simply explain the mechanism you would use to ensure that: (i) the jockey served customers in the order they reserved the space at a free gas pump; (ii) the jockey gets the information about how much fuel is required from the customer; and (iii) the jockey tells each customer when to depart.

4. Recently, the ‘‘leap year virus’’ has been spreading over the Internet and infecting the random number generators on many computers. If your computer is infected with this virus, then whenever you attempt to generate a $U(0,1)$ pseudorandom sequence

$$U_1, U_2, U_3, \cdots$$

the reported value of U_i is forced to be **greater than 0.5** (by reporting its ‘‘true value’’ plus 0.5, if necessary) whenever i is divisible by 4, unless it is divisible by 100 but not by 400. Note that the virus only affects the reported values, and does not change the internal state of the generator.

Describe a method for testing your random number generation that will detect this infection. Explain how you would decide whether the random number generator ‘‘passes’’ or ‘‘fails’’ the test. Don’t forget that half the time an *uninfected* random number generator will *also* produce a value greater than 0.5 under the same conditions.

Formulas:

Poisson Distribution: Number of arrivals during time t with arrival rate λ is (mean= λt, variance= λt).

$$P[k] = \frac{(\lambda t)^k}{k!} \cdot e^{-\lambda t}$$

Chi-Squared test: $\chi^2 = \sum_{j=1}^{k} \frac{(N_j - np_j)^2}{np_j}$

Confidence Intervals: $\bar{X}(n) = \frac{1}{n} \sum_{i=1}^{n} X_i \quad S^2(n) = \frac{1}{(n-1)} \sum_{i=1}^{n} (X_i - \bar{X}(n))^2$

$Prob[abs(\bar{X}(n) - \mu)/\sqrt{\sigma^2/n} < Z_{(1-\alpha/2)}] = 1 - \alpha \qquad Prob[abs(\bar{X}(n) - \mu)/\sqrt{S^2(n)/n} < t_{(n-1),(1-\alpha/2)}] = 1 - \alpha$