IN4390 – Quantitative Evaluation of Embedded Systems
practice exam, 3 hrs

Mitra Nasri, Koen Langendoen, Lydia Chen, Marco Zuniga

Question:	1	2	3	4	5	6	7	Total
Points:	8	15	10	5	10	10	10	68
Score:								

- This is a closed book exam

- You may use a **simple** calculator only (i.e. graphical calculators are not permitted)

- Write your answers with a black or blue pen, not with a pencil

- **Always justify your answers, unless stated otherwise**

- The exam covers the following material:

  (a) the paper "Basic Concepts and Taxonomy of Dependable and Secure Computing" by A. Avizienis ; J.-C. Laprie ; B. Randell ; C. Landwehr

  (b) chapters 18-20,22-23 (DoE), and 30-33 (Queueing Theory) of the book "The Art of Computer Systems Performance Analysis" by R. Jain

  (c) the paper "Petri nets: Properties, analysis and applications" by T. Murata

  (d) chapters 11.2 (DTMC), and 11.3 (CTMC) of the book "Introduction to probability, statistics, and random processes" by H. Pishro-Nik

  (e) the paper "Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration" by A.D. Pimentel

Operational Laws	
Utilization law	$U = XS$
Little's law	$N = XR$
Forced-flow law	$X_k = V_k X$
Bottleneck law	$U_k = D_k X$
Operational Bounds	
Througput	$X \leq \min\left(\dfrac{1}{D_{max}}, \dfrac{N}{D + Z}\right)$
Response time	$R \geq \max\left(D, N \times D_{max} - Z\right)$
Queueing Theory M/M/1	
Utilization	$U = XS = \lambda/\mu = \rho$
Probability of n clients in the system	$P_n = \rho^n(1 - \rho)$
Mean #clients in the system	$N = \rho/(1 - \rho) = \lambda/(\mu - \lambda)$
Mean #clients in the queue	$N_Q = N - \rho$
Mean response time	$R = N/\lambda = 1/(\mu - \lambda)$
Mean waiting time	$W = R - S = \rho/(\mu - \lambda)$
Basic Math	
Geometric series	$\displaystyle\sum_{k=0}^{\infty} r^k = \dfrac{1}{1 - r}, \text{ for } |r| < 1$

ANOVA Table for One Factor Experiments

Component	Sum of Squares	%Variation	DF	Mean Square	F-Comp.	F-Table
y	SSY=$\sum y_{ij}^2$		ar			
$\bar{y}_{..}$	SS0=$ar\mu^2$		1			
y-$\bar{y}_{..}$	SST=SSY-SS0	100	ar-1			
A	SSA = $r\Sigma\, \alpha_i^2$	$100\left(\frac{SSA}{SST}\right)$	a-1	MSA = $\frac{SSA}{a-1}$	$\frac{MSA}{MSE}$	$F_{[1-\alpha;\, a-1,\, a(r-1)]}$
e	SSE=SST- SSA	$100\left(\frac{SSE}{SST}\right)$	$a(r-1)$	MSE = $\frac{SSE}{a(r-1)}$		

Question 1 [8 points]

Answer the following set of true/false questions on the special answer sheet that is machine readable. (Do not forget to fill out your credentials on the answer sheet.)

(a) ☐ 1 point ☐ The 'expected time until next failure' is an example of extrema quantitative measures. **(true/false)**

(b) ☐ 1 point ☐ Latency is a time delay between the cause and the effect of some physical change in the system being observed. **(true/false)**

(c) ☐ 1 point ☐ A dependable system will never occur any failure **(true/false)**

(d) ☐ 1 point ☐ Reliability is the readiness for correct service. **(true/false)**

(e) ☐ 1 point ☐ An incorrect internal state of a system is an error. **(true/false)**

(f) ☐ 1 point ☐ Absence of improper system alterations is not an attribute of dependability, rather it is an attribute of security. **(true/false)**

(g) ☐ 1 point ☐ A point X in the design space of an embedded system is said to Pareto-dominate another point Y if X does better than Y in all (performance/cost) dimensions. **(true/false)**

(h) ☐ 1 point ☐ Multi-objective optimization is a heuristic search method, and can therefore only approximate the true solution (i.e. the Pareto front). **(true/false)**

Question 2 [15 points]

Answer the following short questions.

(a) ☐ 5 points ☐ List all attributes of dependability.

(b) ☐ 4 points ☐ List all means of dependability and security.

(c) ☐ 6 points ☐ Determine the type of "means of dependability and security" for each of the following actions.

(a) debugging the code by local program developers
(b) using a watchdog timer to reset a task when it hangs
(c) following coding standards
(d) using less hazardous programming languages (not like C++)
(e) simplifying the user interface
(f) using three different hardware that run the same program and then use a voter to pick the output that the majority of the computers agree on.

Question 3 [10 points]

A master student conducts a 2-level factorial randomized design. She has then constructed the following incomplete ANOVOA

Source	SS	DF	MS	F
A	350.00	2		
B	300.00		150	
AB	200.00		50	
Error	150.00	18		
Total	1000.00			

(a) 2 points How many levels of factor B did she use in the experiments?

(b) 2 points How many degrees of freedom are associated with the interaction term?

(c) 2 points How many replications has the student performed of each experiment?

(d) 2 points What is the error mean square?

(e) 2 points When the corresponding P values of factor A, B, and AB are 0.02, 0.04, and 0.15, respectively, which factors are significant?

Question 4 [5 points]

Below is a design for measuring the job latency from three replicates based on 5 factors, A, B, C, D, and E.

A	B	C	D	E	latency		
−	−	−	−	−	7.78	7.78	7.81
+	−	−	+	−	8.15	8.18	7.88
−	+	−	+	−	7.50	7.56	7.50
+	+	−	−	−	7.59	7.56	7.75
−	−	+	+	−	7.54	8.00	7.88
+	−	+	−	−	7.69	8.09	8.06
−	+	+	−	−	7.56	7.52	7.44
+	+	+	+	−	7.56	7.81	7.69
−	−	−	−	+	7.50	7.25	7.12
+	−	−	+	+	7.88	7.88	7.44
−	+	−	+	+	7.50	7.56	7.50
+	+	−	−	+	7.63	7.75	7.56
−	−	+	+	+	7.32	7.44	7.44
+	−	+	−	+	7.56	7.69	7.62
−	+	+	−	+	7.18	7.18	7.25
+	+	+	+	+	7.81	7.50	7.59

(a) 4 points Write out the alias structure for this design. In other words, list all confounding factors.

(b) 1 point what is the resolution of the above design?

Question 5 [10 points]

You have designed an embedded system and have run a series of trials to test it. The outcomes of these trials are either success or failure. During your evaluation, you noticed that if the two most recent trials were both successes, the next trial is a success with probability 0.8; otherwise, the chance of success is 0.5. In the long run, what proportion of trials are successes?

Question 6 [10 points]

You have designed an embedded system that has two main components: A and B. The system has three states:

State 1 [S1]: both components are working.

State 2 [S2]: only component A is working.

State 3 [S3]: only component B is working.

There are no self-loops. From S1 you can go to either S2 (component A fails) or S3 (component B fails). From S2 and S3, you can only go back to S1 (i.e. the respective component is repaired).

(a) 4 points (Discrete Time Markov Chains) Assuming the transition probabilities given below, can we obtain a steady-state distribution? If your answer is Yes, provide the steady-state probabilities. If your answer is No, explain why.

S1→S2 = S1→S3 = 0.5
S2→S1 = S3→S1 = 1.0

(b) 6 points (Continuous Time Markov Chains) In CTMCs, the chain does **not** need to be aperiodic to have steady-state probabilities. Considering that aperiodicity is not needed and assuming the rates given below, can we obtain a steady-state distribution? If your answer is Yes, provide the steady-state probabilities. If your answer is No, explain why.

S1→S2 = S1→S3 = λ_1
S2→S1 = S3→S1 = λ_2

Question 7 [10 points]

Operational laws are a crude, but useful tool to do back-of-the-envelope calculations about the performance benefits of adding hardware resources to a computing system. Consider a database server consisting of a CPU and two disks, whose performance characteristics are as follows:

t (measurement time)	650 seconds
C (#completions)	200 queries
B_{CPU} (busy time)	400 seconds
$B_{slow-disk}$ (busy time)	100 seconds
$B_{fast-disk}$ (busy time)	600 seconds
C_{CPU} (#completions)	22,200 jobs
$C_{slow-disk}$ (#completions)	2,000 jobs
$C_{gast-disk}$ (#completions)	20,000 jobs

(a) 3 points Compute the performance benefit of adding a second CPU (i.e., how many more database queries can be handled per second?)

(b) 5 points Compute the maximum performance benefit that can be gained by balancing the load of the slow and fast disk by shifting database files from one to the other.

(c) $\boxed{\text{2 points}}$ Consider the alternative approach of adding a second fast disk. How much can performance be improved this way?