Your name:

CSE 30341 Operating Systems: Module 2 Exam

OPEN BOOK, OPEN NOTES, CLOSED ELECTRONIC SEARCHES

INDIVIDUAL EFFORT

DURATION: 30 MINUTES

There are 6 True of False questions. All questions carry equal weight. If you answer the question correctly, then you get 1 points. If you get it wrong, then you get -1 points. If you don't attempt a particular question, then you get 0 points for that question. The maximum and minimum scores for this exam are +6 and -6, respectively.

Ockham's Razor principles (or Keep It Simple, Stupid: KISS principles) hold. *Make no more assumptions than is needed*; you only have 5 minutes per question. You may write your justification for your answer in the space below. I may look at the argument to give partial credit. You can also use these argument to convince me in person that your answer was correct (using only the assumptions and arguments written by you in the answer). Football penalty rules do not apply to his exam, there will be no penalty for you in trying to convince me

	Question	Answer
1	Modern operating systems such as Windows XP do not detect deadlocks. When some processes enter into a deadlock on these machines, the rest of the system can continue to work normally	
2	Dining philosophers problem will always be deadlock free if we have a free-floating fork in the middle of the table that any philosopher can use (when the forks near them are being used). Philosophers will not wait to pick up a fork that is in use by others	
3	Replacing the atomic transaction's *beginTransaction* and *commitTransaction* with a *binary semaphore* wait() and signal() would provide the same atomic behavior	
4	Consider a system consisting of four processes and a single resource. The current state of the max and allocated is max = [3 2 9 7] and allocated = [1 1 3 2]. The minimum number of units of the resources needed to be available for this state to be safe is 3	
5	Monitors and binary semaphores are always equivalent in terms of synchronization problems that they can solve	
6	Critical section protects shared variables. Multi-thread programs that use no shared variables is always deadlock free	