

CMSC330 Spring 2014 Quiz #3 Solution

1. (12 pts) Prolog

Given the following clauses, list all answers returned by the following queries.

<pre>avenger(thor). avenger(captainAmerica). sibling(thor,loki). asgardian(thor). asgardian(X) :- sibling(Y,X),asgardian(Y). train1(X,Y) :- avenger(X),!,avenger(Y). train2(X,Y) :- avenger(X),X\=Y,avenger(Y).</pre>	<pre>foo([_!T],R) :- foo(T,R). foo([H1,H2!T],H1) :- H1 = H2.</pre>
a. (1 pt) ?- avenger(captainAmerica). true.	f. (2 pt) ?- foo([1,2,3],A). false.
b. (1 pt) ?- asgardian(A). A= thor; A= loki.	g. (2 pts) ?- foo([2,2,4],A). A=2.
c. (1 pt) ?- train1(A,B). A= thor, B= thor; A= thor, B= captainAmerica.	h. (2 pts) ?- foo([5,2,2,3,4,4],A) A=4; A=2.
d. (1 pt) ?- train2(A, captainAmerica). A= thor.	
e. (2 pt) ?- train2(captainAmerica,A). false.	

2. (8 pts) Multithreading

<pre> class Buffer { Buffer () { Object buf = null; boolean empty = true; } </pre>	<pre> void produce(o) { synchronized (this) { 1. if (!empty) wait(); 2. empty = false; 3. notifyAll(); 4. buf = o; } } </pre>	<pre> Object consume() { synchronized (this) { 5. if (empty) wait(); 6. empty = true; 7. notifyAll(); 8. return buf; } } </pre>
---	---	--

Consider the preceding multithreaded Java 1.4 code. Assume there are multiple producer and consumer threads being executed in the program, but only a *single* Buffer object. Questions about the “last statement executed” by two threads x & y refer to the most recently executed statement by those threads at some arbitrary time during the program execution. It does not mean the last statement executed by a thread before the thread exits. If a situation is possible, you need to give an example of how it is possible (e.g., thread x gets to statement a, then thread y gets to statement b). If a situation is not possible, you need to explain why.

- a. (2 pts) Is it possible given two threads x and y for the last statement executed by thread x to be statement 2 and thread y to be statement 7 in the code above? Explain your answer.

No, since if the last statement executed by a thread is either statement 2 or 7, it must be holding the lock. No other thread can acquire the lock and reach statement 2 or 7.

- b. (3 pts) Is it possible given two threads x and y for the last statement executed by thread x to be statement 3 and thread y to be statement 5 in the code above? Explain your answer.

Yes, since thread x may have reached statement 5 and released the lock after calling wait, so thread y may acquire the lock and reach & execute statement 3.

- c. (3 pts) Is it possible in the code above for two threads calling consume() to get the same value from the Buffer object? Explain your answer.

Yes, since the wait at line 5 is not called in a while loop. If there are multiple consumer threads waiting, it is possible for a thread at line 5 to be woken and execute lines 6-8 even though empty is already true. This will cause the thread to return the buffer value already returned by a different consumer thread.