Q17_18

Due Mar 4 at 11:59pm **Points** 5 **Questions** 5 **Time Limit** None
Allowed Attempts 2

Instructions

This quiz is based on content from Sets 17 and 18 on the subject of Thread Synchronization

Attempt History

	Attempt	Time	Score
KEPT	**Attempt 2**	2 minutes	5 out of 5
LATEST	**Attempt 2**	2 minutes	5 out of 5
	Attempt 1	812 minutes	4 out of 5

Score for this attempt: **5** out of 5
Submitted Mar 4 at 9:56pm
This attempt took 2 minutes.

Question 1 — 1 / 1 pts

When a thread T signals a condition variable C,

Correct!

- (●) if other threads are on C's waitlist, one is woken up
- ( ) if other threads are on C's waitlist, all are woken up
- ( ) the thread T is suspended until another thread is added to C's waitlist

Question 2

1 / 1 pts

CV::wait(Lock *lock) releases the lock and starts waiting

- in an automatic operation
- Correct! in an atomic operation
- as two distinct non-atomic sequences of operations

Question 3

1 / 1 pts

A mutex lock can be shared by

- Correct! any number of threads
- exactly two threads

Question 4

1 / 1 pts

To reliably prevent race conditions, threads should

- Correct! use locks
- rely on compiler
- use timers

Question 5

1 / 1 pts

A thread signals a condition variable in order to

Correct!

- (●) notify another thread
- ( ) kill the condition variable
- ( ) put itself on the condition variable's waitlist

Quiz Score: **5** out of 5