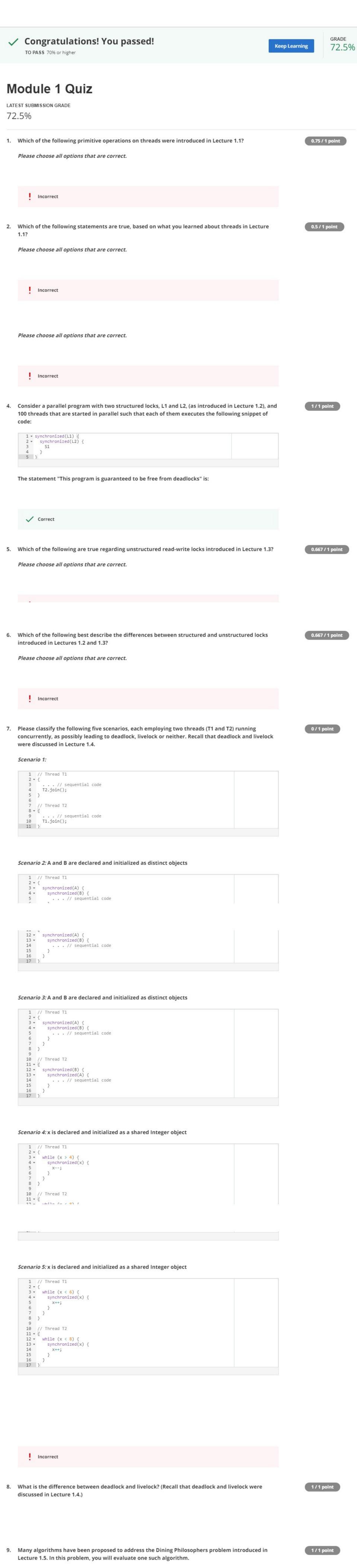
Module 1 Quiz Due Dec 15, 11:59 PM PST



The following simple algorithm can deadlock with all philosophers holding their left chopsticks.

1 // All philosophers:

an odd number of philosophers?

```
2 → while (True) {
3 acquire philosopher's left chopstick
4 acquire philosopher's right chopstick
5
    release right chopstick
```

the chopsticks in the opposite order. More precisely, we propose separate algorithms for the evennumbered and odd-numbered philosophers. Assume the philosophers are numbered 1, ..., n clockwise around the table. 1 // All even-numbered philosophers: 2 - while (True) {

30, we develop a slight variant that attempts to avoid this problem — adjacent philosophiers pick up

```
3 acquire philosopher's left chopstick
 4 acquire philosopher's right chopstick
 5 eat
      release right chopstick
      release left chopstick
 10 // All odd-numbered philosophers:
 11 → while (True) {
 12 acquire philosopher's right chopstick
 13 acquire philosopher's left chopstick
 14
 15
      release left chopstick
 16
      release right chopstick
17 }
```

What liveness issues could arise if we have an even number of philosophers? What about if we had

```
✓ Correct
```

10. Towards the end of Lecture 1.5, Dr. Sarkar mentioned that deadlock and livelock can be avoided in the Dining Philosophers problem by modifying the "all acquire left fork first" algorithm such that n-1 philosophers attempt to acquire their left fork first, and 1 philosopher attempts to acquire its right fork first. However, nothing was mentioned about the impact of this modification on another important liveness issue: starvation. How could starvation occur, or not occur, with this modification presented at the end of Lecture 1.5?

