# Quiz 4

**Due** Jul 30, 2021 at 11:59pm **Points** 100 **Questions** 2
**Available** Jul 29, 2021 at 12am - Jul 30, 2021 at 11:59pm **Time Limit** 60 Minutes

# Instructions

Dear all,

This is the last quiz of this course. This quiz will be available on Canvas from **11:59 pm July. 29 to 11:59 pm July. 30** (**24 hours**, **U.S. Central Time**) so you can take it anytime during this period. Moreover, you have **60 minutes** with only **ONE attempt** to complete and turn in your answers so please make sure you complete all questions before submitting. If there is any problem, please let me know by email.

Good luck!

Your TAs

This quiz is no longer available as the course has been concluded.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| **LATEST** | Attempt 1 | 20 minutes | 100 out of 100 |

Correct answers are hidden.

Score for this quiz: **100** out of 100
Submitted Jul 30, 2021 at 7:17pm
This attempt took 20 minutes.

### Question 1

**50 / 50 pts**

Prove or disprove that a Dining Philosophers solution that permits only four philosophers to be seated at any given time is deadlock-free. Assume philosophers pick up their right chopstick first and then their left, blocking indefinitely to obtain each one.

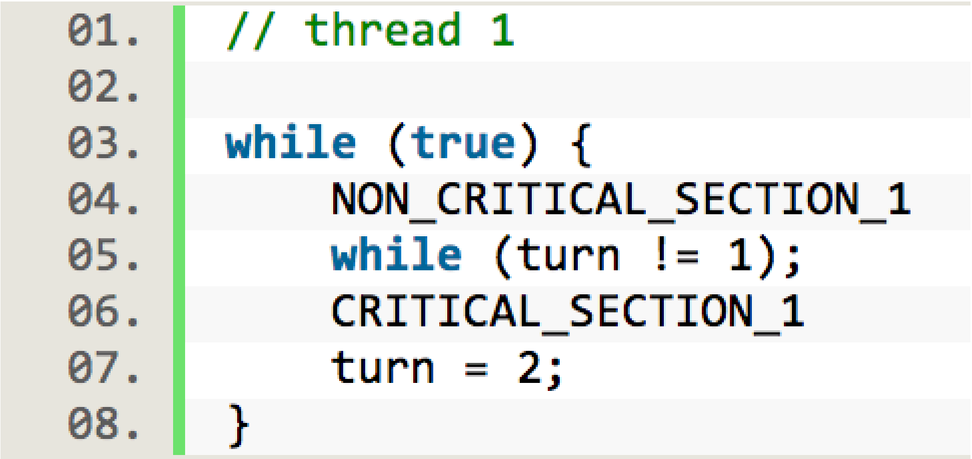
Your Answer:

To prove that this solution is deadlock-free, let's first go back to the original problem. Deadlock occurs when all five of the philosophers pick up their right chopstick simultaneously, never allowing anyone to ever pick up their left chopstick. Now let's consider this situation with only four philosophers at the table at any given time. When all four philosophers pick up their right chopstick at the same time, the first three philosophers clockwise of the empty seat will be let waiting on their left chopstick since it is the right chopstick of their neighbor. However, the final and fourth philosopher will always be able to grab their other chopstick in this situation since their is no one in the seat to their left to grab the left chopstick. So when this philosopher finishes eating it will free up the others one by one. So this solution prevents any deadlock.

## Question 2

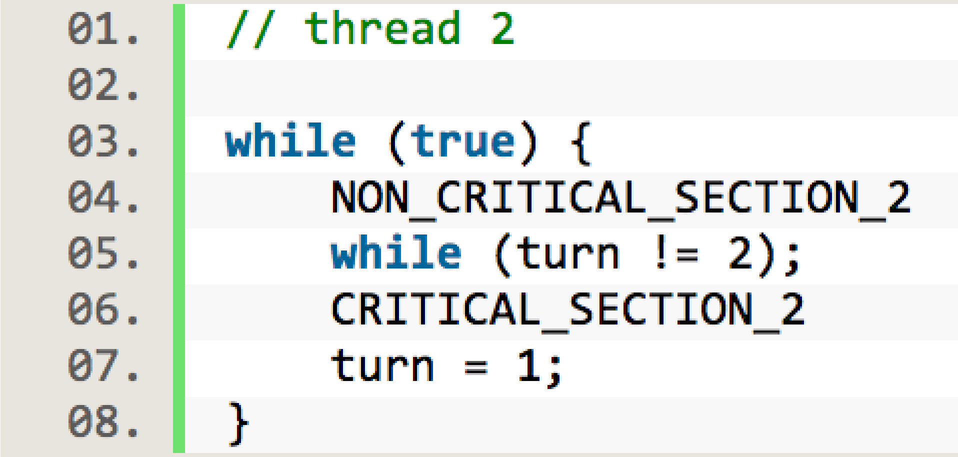
50 / 50 pts

What's wrong with the following attempt to enforce mutual exclusion, besides the fact that it polls?  Here "turn" is a global variable initialized to 1.



```
// thread 1

while (true) {
      NON_CRITICAL_SECTION_1
      while (turn != 1);
      CRITICAL_SECTION_1
      turn = 2;
}
```



```
// thread 2

while (true) {
     NON_CRITICAL_SECTION_2
     while (turn != 2);
     CRITICAL_SECTION_2
     turn = 1;
}
```

Your Answer:

The problem with this attempt at mutual exclusion is that it is not atomic and the two threads could end up running their critical sections at the same time.

Quiz Score: **100** out of 100