Question 1

1. Why Deadlocks Can Occur in the Problem Setup:

The dining philosophers problem is prone to deadlocks due to the potential for circular waiting. In the classic setup, each philosopher must acquire two forks to eat, and if all philosophers simultaneously pick up the fork on their left, a circular dependency occurs, resulting in a deadlock. Additionally, when a philosopher picks up one fork and is unable to acquire the second one, they can lead to resource contention, potentially causing a deadlock.

2. How the Proposed Solution Avoids Deadlock:

In our new look at the problem, we add bowls to the mix as another shared thing. This answer depends on a mix of mutexes (locks) and conditional variables. They make sure that philosophers can only have their meals when they own both forks and a bowl. By bringing in a mutex for the bowls and a conditional variable to tell when a bowl is ready to use, the answer keeps deadlocks from happening. It does this by making sure that philosophers only get forks and bowls when they can. The conditional variables let the philosophers wait until a bowl is ready. This stops them from grabbing forks too early and helps avoid circular problems.

3. Fairness of the Solution:

The suggested plan wants to be fair. It uses a "mutex" for the shared resource (bowls). A "conditional variable" signals when a bowl is free. This means philosophers wait their turn before they try to get forks. However, fairness might still be affected if the philosophers ask for resources out of order. Take a philosopher who is late in getting a bowl. He might have to sit tight for a bit longer. We boost fairness by letting the bowls mutex go once a philosopher gets a bowl. This lets others see if the bowls are free.

Regarding the estimate of how often a philosopher can eat, the fairness mechanisms in place should, in theory, allow all philosophers to have an equal opportunity to eat. However, due to the non-deterministic nature of thread scheduling and the potential for race conditions, it is challenging to provide an exact estimate. In practice, each philosopher should have a reasonable chance to eat over time, assuming a fair scheduling policy and no external interference. Fine-tuning of parameters, such as sleep durations and mutex handling, could further optimize the fairness of the solution.

Screenshot of output attached: