

Dining Philosophers

Group Project – Due April 14

The **dining-philosophers problem** is considered a classic synchronization problem neither because of its practical importance nor because computer scientists dislike philosophers but because it is an example of a large class of concurrency-control problems. It is a simple representation of the need to allocate several resources among several processes in a deadlock-free and starvation-free manner.

Consider five philosophers who spend their lives thinking and eating. The philosophers share a circular table surrounded by five chairs, each belonging to one philosopher. In the center of the table is a bowl of rice, and the table is laid with five single chopsticks. When a philosopher thinks, she does not interact with her colleagues. From time to time, a philosopher gets hungry and tries to pick up the two chopsticks that are closest to her (the chopsticks that are between her and her left and right neighbors). A philosopher may pick up only one chopstick at a time. Obviously, she cannot pick up a chopstick that is already in the hand of a neighbor. When a hungry philosopher has both her chopsticks at the same time, she eats without releasing the chopsticks. When she is finished eating, she puts down both chopsticks and starts thinking again.

From your book 7.1.3 (page 293)

- Requirements

- Groups should preferably be 2 (3 students at most)
- The solution must be uploaded to GitHub and the location of the repository emailed to the professor PRIOR to the presentation
- Each student will grade both themselves and their partners although final grades will be at my discretion.
- The presentation should include an explanation of the group's solution and how it solves the problem
- The presentation should be **10-20** minutes long, and each student must speak during the presentation to describe some part of the problem and/or group's solution.

- Hints

- Start as soon as possible, or you will not finish this project.
- Each philosopher needs to be in a separate thread
- Consider both the possibility of races and deadlocks
- Your book is a great resource
- Make the solution YOUR OWN!

- Grading criteria

- Presentation Content- 50 points
- Speaking Skills – 17 points
- Solution (source code) – 33 points