Semaphores:

- a **semaphore** is a variable or abstract data type that is used for controlling access, by multiple processes, to a common resource in a concurrent system such as a multiprogramming operating system.
- Semaphores which allow an arbitrary resource count are called **counting semaphores**, while semaphores which are restricted to the values 0 and 1 (or locked/unlocked, unavailable/available) are called **binary semaphores**.
- There's a similarity between mutexes and binary semaphores
- A semaphore is somewhat like an integer variable, but is special in that its operations (increment and decrement) are guaranteed to be atomic—you cannot be halfway through incrementing the semaphore and be interrupted and waylaid by another thread trying to do the same thing.
- Semaphores are one of the ways to tackle deadlock and starvation of processes.
- Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when he has both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher. After he finishes eating, he needs to put down both forks so they become available to others. A philosopher can take the fork on his right or the one on his left as they become available, but cannot start eating before getting both of them. This is known as the Dining philosopher problem.
- A simple way to tackle the philosopher problem is to use a mediator who keep a track of all the threads and only once the mediator tells the philosophers to eat will they do so. In this way the philosophers are the threads and the forks are being realised and locked based on a conditional variable. Hence this will make sure that a deadlock or starvation doesn't occur.

References:

https://en.wikipedia.org/wiki/Semaphore_(programming)
https://see.stanford.edu/materials/icsppcs107/23-Concurrency-Examples.pdf