For **Lab Session 4**, we will have lab sheet 5 and 7
Please do study in lab sheet 5 the following:
-volatile keyword
-thread-safe collection
-java.util.concurrent package: Lock interface, ReentrantLock and Semaphore classes.

Exercises:

- App2
- Lab4 application 2, Petri net from figure 4.4 (implement it with ReentrantLock, you can use the method tryLock() to avoid the deadlock) + Class diagram + State machines diagram
- Lab7 application 4, Petri net from figure 7.4 (implement with Semaphore class) + Class diagram + State machines diagram

Lab 5 app 4 is optional, if you solve it, you'll have extra points for it.

https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/locks/Lock.html

Lock (Java Platform SE 7) - Oracle

Lock implementations provide more extensive locking operations than can be obtained using synchronized methods and statements. They allow more flexible structuring, may have quite different properties, and may support multiple associated Condition objects. A lock is a tool for controlling access to a shared resource by multiple threads.

docs.oracle.com