

Eg 1) Consumer is the 1st process to enter into the monitor. It enters and keep the lock with it since it is the first and nobody holds the lock.  
If a producer tries to enter, it cannot since the key is with the consumer. So the producer will wait in the queue until consumer releases the lock. Since in monitors only 1 process can access the shared variable at a particular time.

Consumer is the 1st process to enter into the monitor. It enters and keep the lock with it since it is the first and nobody holds the lock. As it is the 1st consumer process, there are no items to consume. So for the consumer there are no items to consume, but it enters the critical section. So lock 🡪 doesn’t fails here. Now the condition variables comes into picture

Wait() and signal() on semaphores are entirely different from condition variable’s wait() and signal  
**x.wait()** 🡪 When a process (consumer) knows it cannot proceed further it will become inactive/suspended. So that process releases the lock for the other process(producer) to enter.

x.signal() 🡪 The completed process(producer) will wake up the process(consumer) which is waiting on that condition variable.  
So the producer after producing, it will signal the consumer which is waiting on that condition variable