

**semaphore** : a synchronization object that maintains a count between zero and a specified maximum value. The count is decremented each time a thread completes a wait for the semaphore object and incremented each time a thread releases the semaphore. When the count reaches zero, no more threads can successfully wait for the semaphore object state to become signaled.

**concurrent hash map** : A hash table supporting full concurrency of retrievals and adjustable expected concurrency for updates.

**copy on write arrays** : `CopyOnWriteArrayList` behaves as a `List` implementation that allows multiple concurrent reads, and for reads to occur concurrently with a write. The way it does this is to make a brand new copy of the list every time it is altered.

**Barrier** : a type of synchronization method. A barrier for a group of threads or processes in the source code means any thread/process must stop at this point and cannot proceed until all other threads/processes reach this barrier