**Starvation and Fairness**

* [Causes of Starvation in Java](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#starvation)
  + [Threads with high priority swallow all CPU time from threads with lower priority](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#high-priority)
  + [Threads are blocked indefinitely waiting to enter a synchronized block](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#synchronized)
  + [Threads waiting on an object (called wait() on it) remain waiting indefinitely](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#wait)
* [Implementing Fairness in Java](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#fairness)
  + [Using Locks Instead of Synchronized Blocks](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#locks)
  + [A Fair Lock](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#fairlock)
  + [A Note on Performance](http://tutorials.jenkov.com/java-concurrency/starvation-and-fairness.html#performance)

## ReentrantLock accepts an optional fairness parameter. When set true, under contention, locks favor granting access to the longest-waiting thread. Otherwise this lock does not guarantee any particular access order. Programs using fair locks accessed by many threads may display lower overall throughput (i.e., are slower; often much slower) than those using the default setting, but have smaller variances in times to obtain locks and guarantee lack of starvation. Note however, that fairness of locks does not guarantee fairness of thread scheduling. Thus, one of many threads using a fair lock may obtain it multiple times in succession while other active threads are not progressing and not currently holding the lock. Also note that the untimed [tryLock](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/locks/ReentrantLock.html" \l "tryLock()) method does not honor the fairness setting. It will succeed if the lock is available even if other threads are waiting.