# Java Concurrent Collections: Introduction



Douglas C. Schmidt <u>d.schmidt@vanderbilt.edu</u> www.dre.vanderbilt.edu/~schmidt

> Institute for Software Integrated Systems Vanderbilt University Nashville, Tennessee, USA



#### Learning Objectives in this Lesson



## Learning Objectives in this Lesson

- Understand the capabilities of Java's concurrent collections
  - As well as how Java's concurrent collections overcome limitations with Java's synchronized collections



 Java concurrent collections provide features that are optimized for the needs of concurrent programs These are the concurrent-aware interfaces:

BlockingQueue TransferQueue BlockingDeque ConcurrentMap ConcurrentNavigableMap

Concurrent-aware classes include

LinkedBlockingQueue ArrayBlockingQueue PriorityBlockingQueue DelayQueue SynchronousQueue LinkedBlockingDeque LinkedTransferQueue CopyOnWriteArrayList CopyOnWriteArraySet ConcurrentHashMap

See <u>docs.oracle.com/javase/tutorial/essential/concurrency/collections.html</u>

- Java concurrent collections provide features that are optimized for the needs of concurrent programs
  - A concurrent collection is threadsafe, but is not governed by only a single exclusion lock



See docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-summary.html

- Java concurrent collections provide features that are optimized for the needs of concurrent programs
  - A concurrent collection is threadsafe, but is not governed by only a single exclusion lock
  - They avoid *memory consistency errors* by defining a "happensbefore" relationship



This relationship is a guarantee that memory writes in one thread are visible when read by other threads.

See en.wikipedia.org/wiki/Happened-before

- Java concurrent collections provide features that are optimized for the needs of concurrent programs
  - A concurrent collection is threadsafe, but is not governed by only a single exclusion lock
  - They avoid *memory consistency errors* by defining a "happensbefore" relationship
    - e.g., between a thread that adds an object to a collection with later thread(s) that access or remove that object



See <a href="https://docs.oracle.com/javase/tutorial/essential/concurrency/memconsist.html">docs.oracle.com/javase/tutorial/essential/concurrency/memconsist.html</a>

- Java concurrent collections provide features that are optimized for the needs of concurrent programs
  - A concurrent collection is threadsafe, but is not governed by only a single exclusion lock
  - They avoid *memory consistency errors* by defining a "happensbefore" relationship
  - They enable the desired behavior on blocking queues that are empty or full



See tutorials.jenkov.com/java-util-concurrent/blockingqueue.html

# End of Java Concurrent Collections: Introduction