Java Concurrent Collections:
Introduction

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

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Lesson

• Understand the capabilities of Java’s concurrent collections
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
Overview of Java
Concurrent Collections
Overview of Java Concurrent Collections

• Java concurrent collections provide features that are optimized for the needs of concurrent programs

These are the concurrent-aware interfaces:

- BlockingQueue
- TransferQueue
- BlockingDeque
- ConcurrentHashMap
- ConcurrentNavigableMap

Concurrent-aware classes include:

- LinkedBlockingQueue
- ArrayBlockingQueue
- PriorityBlockingQueue
- DelayQueue
- SynchronousQueue
- LinkedBlockingDeque
- LinkedTransferQueue
- CopyOnWriteArrayList
- CopyOnWriteArraySet
- ConcurrentHashMap

See docs.oracle.com/javase/tutorial/essential/concurrency/collections.html
Overview of Java Concurrent Collections

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

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-summary.html
Overview of Java Concurrent Collections

- Java concurrent collections provide features that are optimized for the needs of concurrent programs
  - A concurrent collection is thread-safe, but is not governed by only a single exclusion lock
  - They avoid *memory consistency errors* by defining a “happens-before” relationship

See [en.wikipedia.org/wiki/Happened-before](en.wikipedia.org/wiki/Happened-before)
Overview of Java Concurrent Collections

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

See docs.oracle.com/javase/tutorial/essential/concurrency/memconsist.html
Overview of Java Concurrent Collections

- Java concurrent collections provide features that are optimized for the needs of concurrent programs
  - A concurrent collection is thread-safe, but is not governed by only a single exclusion lock
  - They avoid *memory consistency errors* by defining a “happens-before” 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