### Learning Objectives in this Part of the Lesson

- Be aware of the Java memory model
- Understand the purpose of Java synchronizers
- Recognize the pervasiveness of Java synchronizers
- Know the types of capabilities provided by Java synchronizers

<table>
<thead>
<tr>
<th>Category</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic operations</td>
<td>An action that effectively happens all at once or not at all</td>
</tr>
<tr>
<td>Mutual exclusion</td>
<td>Allows concurrent access &amp; updates to shared mutable data without race conditions</td>
</tr>
<tr>
<td>Coordination</td>
<td>Ensures computations run properly, e.g., in the right order, at the right time, under the right conditions, etc.</td>
</tr>
<tr>
<td>Barrier synchronization</td>
<td>Ensures that any thread(s) must stop at a certain point &amp; cannot proceed until all other thread(s) reach this barrier</td>
</tr>
<tr>
<td>Category</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Atomic operations</td>
<td>An action that effectively happens all at once or not at all</td>
</tr>
<tr>
<td>Mutual exclusion</td>
<td>Allows concurrent access &amp; updates to shared mutable data without race conditions</td>
</tr>
<tr>
<td>Coordination</td>
<td>Ensures computations run properly, e.g., in the right order, at the right time, under the right conditions, etc.</td>
</tr>
<tr>
<td>Barrier synchronization</td>
<td>Ensures that any thread(s) must stop at a certain point &amp; cannot proceed until all other thread(s) reach this barrier</td>
</tr>
</tbody>
</table>
# Learning Objectives in this Part of the Lesson

- Be aware of the Java memory model
- Understand the purpose of Java synchronizers
- Recognize the pervasiveness of Java synchronizers
- Know the types of capabilities provided by Java synchronizers

<table>
<thead>
<tr>
<th>Category</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic operations</td>
<td>An action that effectively happens all at once or not at all</td>
</tr>
<tr>
<td>Mutual exclusion</td>
<td>Allows concurrent access &amp; updates to shared mutable data without race conditions</td>
</tr>
<tr>
<td>Coordination</td>
<td>Ensures computations run properly, e.g., in the right order, at the right time, under the right conditions, etc.</td>
</tr>
<tr>
<td>Barrier synchronization</td>
<td>Ensures that any thread(s) must stop at a certain point &amp; cannot proceed until all other thread(s) reach this barrier</td>
</tr>
</tbody>
</table>
Types of Java Synchronizer Capabilities
Types of Java Synchronizer Capabilities

- Java synchronizers provide various types of capabilities, e.g.
  - Atomic ordering
  - Mutual exclusion
  - **Coordination**
    - Ensures computations run properly
Java synchronizers provide various types of capabilities, e.g.

- Atomic ordering
- Mutual exclusion
- Coordination
  - Ensures computations run properly, e.g.
  - In the right order

Types of Java Synchronizer Capabilities

See github.com/douglascraigschmidt/LiveLessons/tree/master/PingPongApplication
Types of Java Synchronizer Capabilities

- Java synchronizers provide various types of capabilities, e.g.
  - Atomic ordering
  - Mutual exclusion
  - **Coordination**
    - Ensures computations run properly, e.g.
      - In the right order
      - At the right time

See [en.wikipedia.org/wiki/Real-time_computing](en.wikipedia.org/wiki/Real-time_computing)
Types of Java Synchronizer Capabilities

- Java synchronizers provide various types of capabilities, e.g.
  - Atomic ordering
  - Mutual exclusion
  - Coordination
    - Ensures computations run properly, e.g.
      - In the right order
      - At the right time
      - Under the right conditions

See [github.com/douglascraigschmidt/LiveLessons/tree/master/PalantiriManagerApplication](https://github.com/douglascraigschmidt/LiveLessons/tree/master/PalantiriManagerApplication)
Java synchronizers provide various types of capabilities, e.g.

- **Atomic ordering**
- **Mutual exclusion**
- **Coordination**
  - Ensures computations run properly
- Coordination is supported by the Java concurrent & locks packages
  - e.g., ConditionObject, Semaphore, etc.

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-summary.html](http://docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-summary.html)
Java synchronizers provide various types of capabilities, e.g.

- Atomic ordering
- Mutual exclusion
- **Coordination**
  - Ensures computations run properly
  - Coordination is supported by the Java concurrent & locks packages
  - Coordination is also supported by Java built-in monitor objects

Java synchronizers provide various types of capabilities, e.g.

- Atomic ordering
- Mutual exclusion
- Coordination
- **Barrier synchronization**
  - Ensures that any thread(s) must stop at a certain point & cannot proceed until all thread(s) reach the barrier

**Barrier synchronization** is a variant of coordination
Types of Java Synchronizer Capabilities

- Java synchronizers provide various types of capabilities, e.g.
  - Atomic ordering
  - Mutual exclusion
  - Coordination
  - **Barrier synchronization**
    - Ensures that any thread(s) must stop at a certain point & cannot proceed until all thread(s) reach the barrier
  - Barrier synchronization is supported by the Java concurrent package
    - e.g., CountDownLatch, CyclicBarrier, Phaser, etc.

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/package-summary.html
Java synchronizers provide various types of capabilities, e.g.

- **Atomic ordering**
- **Mutual exclusion**
- **Coordination**
- **Barrier synchronization**
  - Ensures that any thread(s) must stop at a certain point & cannot proceed until all thread(s) reach the barrier
  - Barrier synchronization is supported by the Java concurrent package
  - Barrier synchronization is also supported by the Thread.join() method

See [docs.oracle.com/javase/8/docs/api/java/lang/Thread.html#join](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html#join)
We’ll cover all these types of Java synchronizers in this course!!

<table>
<thead>
<tr>
<th>Category</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic operations</td>
<td>An action that effectively happens all at once or not at all</td>
</tr>
<tr>
<td>Mutual exclusion</td>
<td>Allows concurrent access &amp; updates to shared mutable data without race conditions</td>
</tr>
<tr>
<td>Coordination</td>
<td>Ensures computations run properly, e.g., in the right order, at the right time, under the right conditions, etc.</td>
</tr>
<tr>
<td>Barrier synchronization</td>
<td>Ensures that any thread(s) must stop at a certain point &amp; cannot proceed until all other thread(s) reach this barrier</td>
</tr>
</tbody>
</table>
End of Types of Java Synchronizer Capabilities (Part 2)