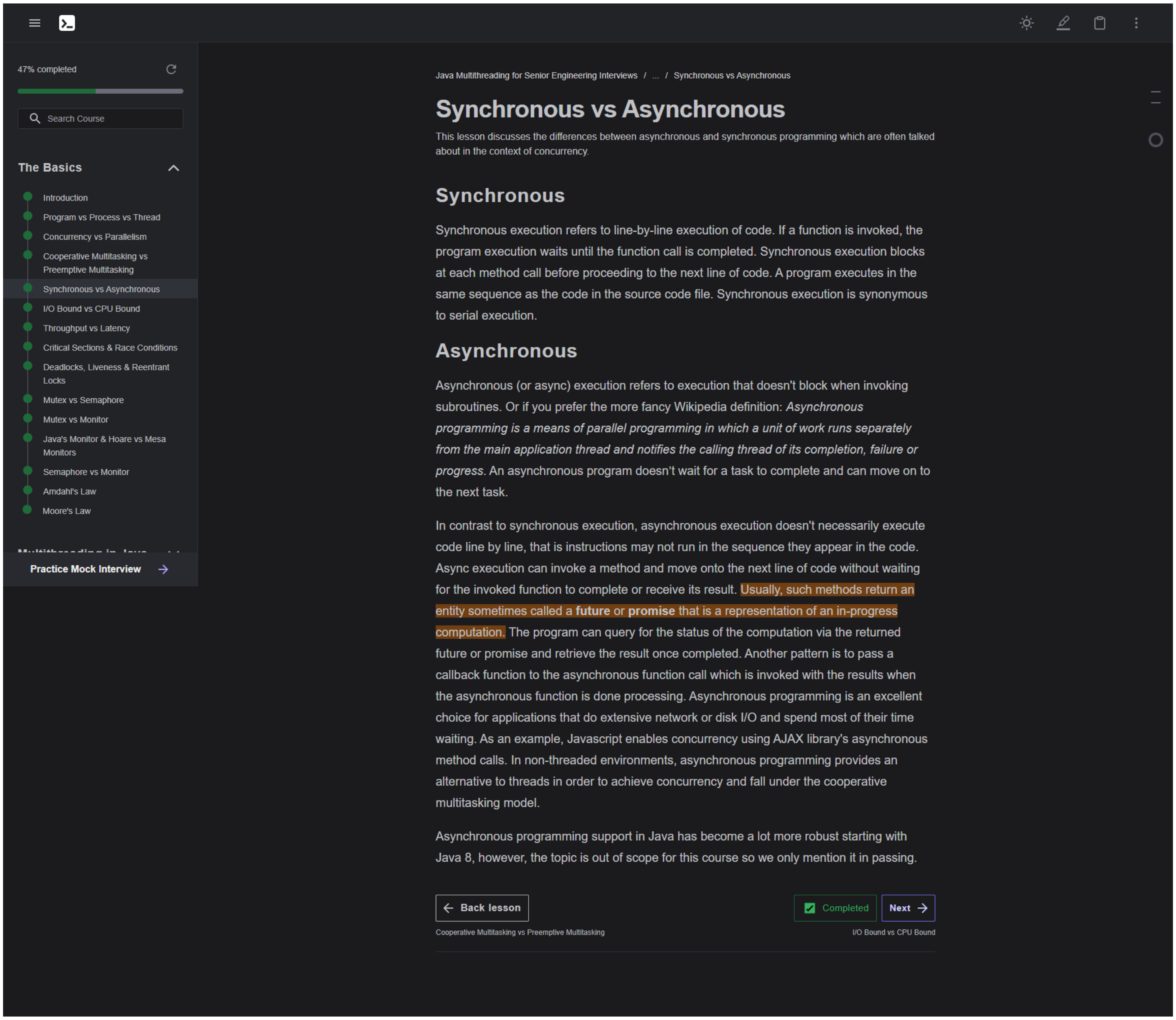

Synchronous vs Asynchronous

This lesson discusses the differences between asynchronous and synchronous programming which are often talked about in the context of concurrency.

Synchronous

Synchronous execution refers to line-by-line execution of code. If a function is invoked, the program execution waits until the function call is completed. Synchronous execution blocks at each method call before proceeding to the next line of code. A program executes in the same sequence as the code in the source code file. Synchronous execution is synonymous to serial execution.

Asynchronous

Asynchronous (or async) execution refers to execution that doesn't block when invoking subroutines. Or if you prefer the more fancy Wikipedia definition: *Asynchronous programming is a means of parallel programming in which a unit of work runs separately from the main application thread and notifies the calling thread of its completion, failure or progress*. An asynchronous program doesn't wait for a task to complete and can move on to the next task.

In contrast to synchronous execution, asynchronous execution doesn't necessarily execute code line by line, that is instructions may not run in the sequence they appear in the code. Async execution can invoke a method and move onto the next line of code without waiting for the invoked function to complete or receive its result. Usually, such methods return an entity sometimes called a **future** or **promise** that is a representation of an in-progress computation. The program can query for the status of the computation via the returned future or promise and retrieve the result once completed. Another pattern is to pass a callback function to the asynchronous function call which is invoked with the results when the asynchronous function is done processing. Asynchronous programming is an excellent choice for applications that do extensive network or disk I/O and spend most of their time waiting. As an example, Javascript enables concurrency using AJAX library's asynchronous method calls. In non-threaded environments, asynchronous programming provides an alternative to threads in order to achieve concurrency and fall under the cooperative multitasking model.

Asynchronous programming support in Java has become a lot more robust starting with Java 8, however, the topic is out of scope for this course so we only mention it in passing.

Back lesson
Cooperative Multitasking vs Preemptive Multitasking

✓ Completed Next
I/O Bound vs CPU Bound