codecademy

Synchronization

Critical Section

Threads or processes may require shared access to certain resources. Areas of the program where these resources exist are called critical sections.

Synchronization

Synchronizing a concurrent program guarantees that each critical section may only be accessed by at most one thread at a time.

Race Conditions

A race condition exists within a concurrent program when the behavior of that program is dependent on the non-deterministic sequence of operations.

Mutex

A mutual exclusion lock, or mutex, is the mechanism which ensures there may only be one thread inside of a critical section at the same time.

Atomic Operation

Atomic operations are isolated from and independent of all other operations; that is, no thread will ever encounter a partially-completed atomic operation.

Atomic Variable

An atomic variable is a variable whose modification is inherently thread-safe because modifying it takes place as a single atomic operation.

Deadlock

A deadlock exists when a thread requests a lock on a shared resource that, programmatically, will never receive it because another thread possesses the lock and will not release it.

Condition Variable

Condition variables will notify threads when it is time for them to execute rather than continuously check for certain conditions to arise before executing.