Instance Method Synchronization

When a thread enters an instance method synchronization, it acquires a lock on that specific object (instance), which leads to the following behaviours:

* Blocking on the Same Object:
* Other threads cannot enter any other instance synchronized method or block synchronized code on the same object.
* These threads must wait until the lock on that object is released.
* No Blocking on Static Synchronization:
* A thread holding the instance lock does not block other threads from entering static synchronized methods on the same object or class. This is because static methods are synchronized on the class-level lock, which is separate from the object instance lock.
* No Blocking on Different Objects:
* The instance lock is specific to the object. Therefore, other threads can still access instance synchronized methods or synchronized blocks on different instances of the same class (i.e., different objects), as each object has its own independent lock.



Static method synchronization

* If one thread enters a static synchronized method, it will block other threads from entering any static synchronized method in the same class. However, other threads can still access instance synchronized methods or synchronized blocks on any instance of the class, as these locks are independent of the static synchronization lock.



Block synchronization

When a thread enters a block synchronization on an object, it acquires a lock on that specific object (instance). This leads to the following behaviours:

Blocking on the Same Object

* If a thread enters a synchronized block on an object, it locks that object. As a result, other threads cannot enter any instance synchronized methods or other synchronized blocks on the same object.
* These threads must wait until the lock on the object is released.

No Blocking on Static Synchronization

* The instance-level lock acquired by entering the synchronized block does not block other threads from accessing static synchronized methods. This is because static methods are synchronized on the class-level lock, which is independent of the object instance lock.

No Blocking on Different Objects

* Since the lock is specific to the particular object, other threads can still access instance synchronized methods or synchronized blocks on different instances (i.e., different objects of the same class), as each object has its own separate lock.





