**What is meant by the term "synchronization"?**

In Java, synchronization means that a method is synchronized or a statement is synchronized.

You synchronize the method or statement by adding the keyword to the method declaration or statement.

By synchronizing the methods, only one thread can execute the method at a time. Other threads wait until a thread finishes.

By synchronizing a statement, threads are forced to execute a block of code sequentially. Synchronized statements must specify the object that provides the intrinsic lock.

Instead of explicitly setting and releasing locks you perform these operations implicitly by calling a method declared to be synchronized or by executing a block that synchronizes on the object

**When is it necessary to perform synchronization?**

Synchronization is necessary when there are multiple threads to prevent memory errors and interference from other threads.

The following is from: [https://docs.oracle.com/javase/tutorial/essential/concurrency/syncmeth.html (Links to an external site.)](https://docs.oracle.com/javase/tutorial/essential/concurrency/syncmeth.html)

*"if an object is visible to more than one thread, all reads or writes to that*   
*object's variables are done through synchronized methods. (An important exception: final fields, which cannot be modified after the object is constructed, can be safely read through non-synchronized methods, once the object is constructed) This strategy is effective, but can present problems with*[*liveness (Links to an external site.)*](https://docs.oracle.com/javase/tutorial/essential/concurrency/liveness.html)*"*

**Describe a unique (different from classmate) scenario**

**where synchronization would be applicable.**

A scenario where synchronization would be applicable is for a method that is used by multiple threads, such as a counter. By synchronizing the method,

* Only one thread can run the method at a time, preventing inaccurate count data and results
* Changes to the state of the object are visible to all threads.

**Example Code:**

// synchronized method

public class SynchronizedCounter {

    private int count = 0;

    public synchronized void increment() {        count++;    }

    public synchronized void decrement() {        count--;    }

    public synchronized int value() {        return count;    }

}

Another Scenario:

Example code for a synchronized address method:

Instead of using (could give inconsistent data) :

printMyMailingLabel( address.getMyStreetAddress(), address.getMyZipCode() );

Use :

public synchronized MyAddressData getMyAddress()

{

    return new MyAddressData( this.myStreetAddress, this.myZipCode );

}

this example is based an example from: <http://www.cs101.org/ipij/synchronization.html>

It helped to understand the synchronization concepts.