



**Method overloading** means that a class has more than one method with the same name but different parameters (different type or number of arguments).

**1. sleep() Method:**

* **Yes, it is overloaded.**
* The sleep() method in the Thread class is overloaded, meaning there are multiple versions of the method with different parameter lists.
  + **Method 1**: Thread.sleep(long millis)  
    Pauses the execution of the current thread for the specified number of milliseconds.
  + **Method 2**: Thread.sleep(long millis, int nanos)  
    Pauses the execution for the specified milliseconds and additional nanoseconds (for finer control).

**Why is it marked as overloaded?**

Because sleep() has two versions: one that takes only milliseconds and another that takes both milliseconds and nanoseconds.

**2. yield() Method:**

* **No, it is not overloaded.**
* The yield() method is not overloaded. There is only one version of the method:
  + **Method**: Thread.yield()  
    It suggests that the current thread is willing to yield execution to other threads. It does not take any arguments.

**3. join() Method:**

* **Yes, it is overloaded.**
* The join() method is overloaded with three versions:
  + **Method 1**: join()  
    Causes the current thread to wait until the thread on which join() was called finishes execution.
  + **Method 2**: join(long millis)  
    Causes the current thread to wait for the specified number of milliseconds for the thread to finish.
  + **Method 3**: join(long millis, int nanos)  
    Causes the current thread to wait for the specified milliseconds and nanoseconds.