Visibility problem

A [visibility problem](https://www.baeldungtest.com/java-volatile) is one of the issues when working in a multithreaded application. The visibility problem is tightly connected to the Java [memory model](https://www.baeldungtest.com/java-volatile#shared-multiprocessor-architecture).

In multithreaded applications, each thread has its cached version of shared resources and updates the values in or from the main memory based on events or a schedule.

**The thread cache and main memory values might differ.** Therefore, even if one thread updates the values in the main memory, these changes are not instantly visible to other threads. This is called a visibility problem.

**The volatile keyword**[**helps**](https://www.baeldungtest.com/java-volatile-variables-thread-safety)**us to resolve this issue by bypassing caching in a local thread.** Thus, volatile variables are visible to all the threads, and all these threads will see the same value. Hence, when one thread updates the value, all the threads will see the new value.