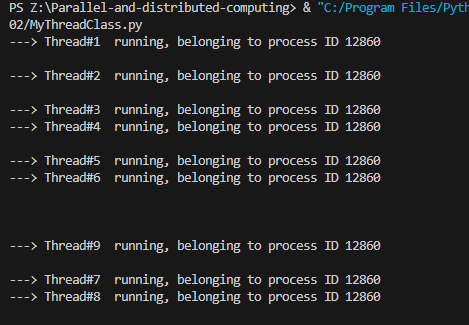
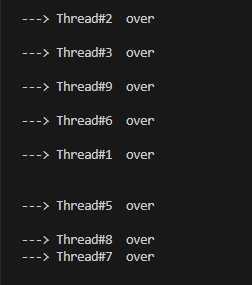
Threads Syncronization

1st code (No Locks):

This code creates 9 threads, and each one waits (sleeps) for a random time between 1 and 10 seconds. All threads are started at the same time, and there is **no lock used** to control them. That means they all run **together at the same time** (concurrently), without waiting for each other. This shows how multithreading works when threads are free to run on their own.

OUTPUT:

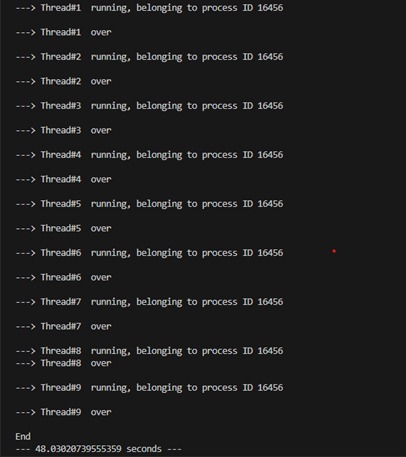




2nd code (Lock around full body):

In this version, a **lock is used for the whole thread work**. Each thread takes the lock before printing and sleeping, and only then the next thread can run. This means the threads run **one after another**, not at the same time. Even though many threads are created, the lock makes them behave like they are running one by one.

OUTPUT:



3rd Code (Lock around only print statement):

In the third code, the **lock is used only for the print part**, not the whole thread. Each thread locks the print statement, prints its message, then unlocks and goes to sleep. This way, threads can still run **at the same time**, but their print messages won’t mix up. It keeps the output clean while still allowing some parallel work.

OUTPUT:

