

Nikita Persikov
ASTE 404
HW 12

Problem 1:

Part c:

A situation that meets the race condition is one where the order of completion of running threads' tasks affects the final result, making the result of the same code not repeatable.

Atomic operations are those which cannot be interrupted by other operations, similar to how atoms are so small that they cannot (with irrelevant caveats) be broken up.

C++ operations are assumed non-atomic unless the compiler specifies otherwise. This means that there is a chance operations happen at the same time on the same memory, making the change appear to have happened once. In this example, this happens when doWork() updates the counter variable twice at the same time. Instead of doing $0+1+1$, it does $0+1$, which is why sometimes asterisks get printed and the code does not run correctly. Thus, because operations are non-atomic, the race condition prevents repeatable results.

Problem 2:

Part 1:

Serial run time of the julia set: **17.682s**

Part 2:

Parallelized (on the inner j dimension) run time of the julia set: **9.67038s**

Part 3:

My computer supports up to 12 threads. The total run time for the code was **5.78801s**.

Therefore, the parallel efficiency is $17.682/(12*5.78801) =$ **0.2546**

