std::atomic<>

- Ensure “tear free” & synchronised manipulation of shared data
- May use traditional locks if data-type cannot be manipulated atomically in hardware. Always check **`std::atomic<>::is_always_lock_free`**!
- Only a subset of manipulations are supported:
  - Store
  - Load
  - Atomic addition/subtraction
  - exchange/compare-exchange
- More info here: https://herbsutter.com/2013/02/11/atomic-weapons-the-c-memory-model-and-modern-hardware/

std::memory_order

- Can be specified to relax the memory ordering of atomic operations
- If you don’t care about the ordering of operations, you might get better performance using **`memory_order_relaxed`**
- More info here: https://herbsutter.com/2013/02/11/atomic-weapons-the-c-memory-model-and-modern-hardware/