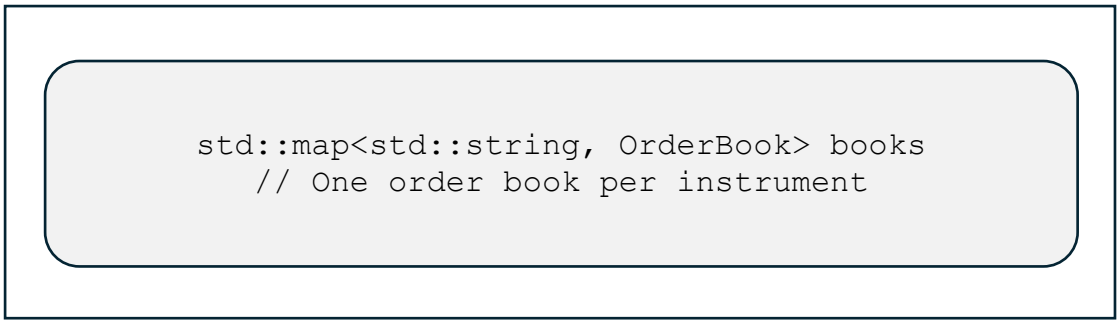
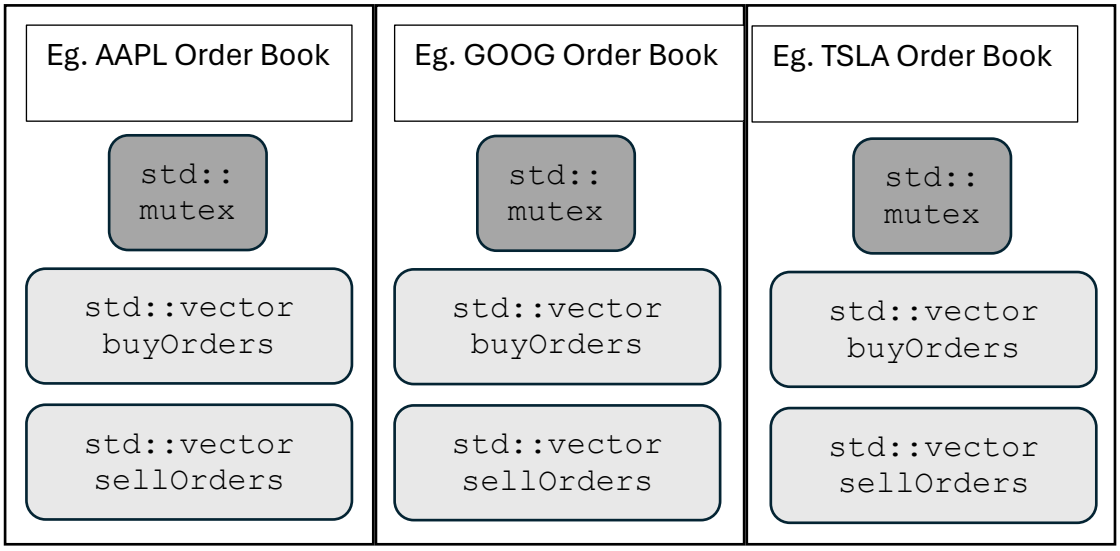
CS3211 Assignment 1

Data Structures

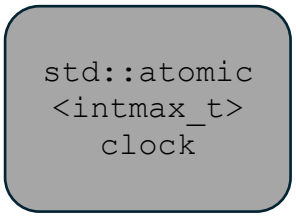
Engine



Order Books



Clock



Engine: The engine stores multiple order books, one for each instrument. The engine uses a `map` to keep track of the `OrderBook` for each instrument. There are no locks on this `map`.

Order Book: The order book utilizes two separate `std::vector<Order>` to store buy and sell orders. A comparator where buy orders are sorted in descending order of

price for priority access to the highest bid and sell orders are sorted in ascending order of price to prioritize the lowest ask is also implemented for the `Order` class. When orders are inserted into the vector, it is inserted in ascending order of the comparator. Each order book has its own mutex, which is locked by the engine before read/write operations are performed. As locks are specific to each order book, which are specific to each instrument, orders for different instruments can execute concurrently.

Clock: An atomic `std::atomic<intmax_t>` is used to keep track of the timestamps. This is automatically incremented when the current time is fetched. This is used to ensure that all times in the program are unique and sequential.

Synchronisation primitives used

Mutexes: An `std::mutex` is employed to protect access to buy and sell order `std::vector`s within the `OrderBook`, ensuring that only one thread can modify the state of the order book at any given time. During access, the `Engine` uses an `std::unique_lock` to lock the mutex.

Atomics: Atomics are used to store generate the timestamp for events (eg. Order added to order book, order execution, order cancellation) to minimize the amount of locks and improve concurrency.

Level of concurrency

The level of concurrency is instrument-level. As each order book is specific to an instrument, and only the mutex for the relevant order book is locked to process the order,

Testing methodology

Mostly handwritten test cases were used with the grader binary for testing. In the interests of time, the provided test cases were extended to a few large comprehensive test cases by incorporating the following:

1. Multiple instruments
2. Multiple threads
3. Orders at different prices
4. Multiple orders at the same price that are differentiated only by time