Integer Operations and Threads Solutions

Integer Operation Example

- Write a program which runs the task function shown on the next slide as 10 concurrent threads

Integer Operations and Threads

```
int counter {0};

void task() {
    for (int i = 0; i < 100'000; ++i)
        ++counter;
}
```

Integer Operation Example

- Briefly explain your results
  - A data race occurs, because multiple threads access the shared variable, at least one thread modifies it, and there is no synchronization between the thread accesses
  - Incrementing an integer is an uninterruptible operation. However, incrementing the counter variable also involves fetching and storing its value. Other threads can interleave between these operations
  - Another thread could change the value after this thread has fetched it. Our thread will use a stale value in its calculation
  - Another thread could publish its value before this thread does. Our thread will publish a stale value to other threads

Thread Synchronization

- Briefly explain how using a mutex avoids these problems
  - Calling lock() prevents other threads from changing the value of counter after this thread has fetched it
  - Calling unlock() publishes the new value and allows other threads to access it
- Rewrite your program to use a mutex