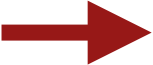
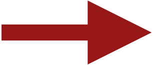
```cpp
class spinlock
{
    std::atomic_flag _flag;
public:
    spinlock() : _flag(ATOMIC_FLAG_INIT)
    {}

    spinlock(const spinlock&) = delete;
    spinlock& operator=(const spinlock&) = delete;
    spinlock(spinlock&&) = default;

    void lock()
    {
        while(_flag.test_and_set(std::memory_order_acquire));
    }

    void unlock()
    {
        _flag.clear(std::memory_order_release);
    }
};
```





```
class spinlock
{
    std::atomic_flag _flag;
public:
    spinlock() : _flag(ATOMIC_FLAG_INIT)
    {}

    spinlock(const spinlock&) = delete;
    spinlock& operator=(const spinlock&) = delete;
    spinlock(spinlock&&) = default;

    void lock()
    {
        while(_flag.test_and_set(std::memory_order_acquire));
    }

    void unlock()
    {
        _flag.clear(std::memory_order_release);
    }
};
```

```
void th_write(spinlock& spin, int& a )
{
    spinlock_guard lk{spin};
    a = 42;
}

void th_read(spinlock& spin, int& a)
{
    spinlock_guard lk{spin};
    std::cout << " value = " << a << "\n";
}
```