

# C++ static code analysis:

## "std::scoped\_lock" should be created with constructor arguments

1 minute

---

When constructing an `std::scoped_lock`, the constructor arguments are used to list the mutexes that the `scoped_lock` will lock on the creation and unlock on destruction. It is possible to construct a `scoped_lock` without any parameter, but in that case, it does absolutely nothing and is just dead code, which was probably not the intent of the user.

### Noncompliant Code Example

```
void f1(std::mutex &m) {  
    std::scoped_lock lock; // Noncompliant  
    // Do some work  
}
```

### Compliant Solution

```
void f1(std::mutex &m) {
```

```
std::scoped_lock lock {m}; // Compliant
// Do some work
}
```

```
template<class... D>
void processAll(D &...data) {
    scoped_lock lock {data.getMutex()...}; // Compliant, even
    if the list might be empty in some cases
    // Do some work
}
```