Implicit **`mutex`** locking

65

```cpp
class(mutex) person
{
public:
    person() = default;

    std::string get_first_name() const
    {
        return first_name;
    }

    void set_first_name (std::string_view new_first)
    {
        first_name = new_first;
    }

    // Repeat for last_name

private:
    std::string first_name, last_name;
};
```

```cpp
class person
{
public:
    person() = default;

    std::string get_first_name() const
    {
        std::scoped_lock _ (mutex);
        return person_.get_first_name();
    }

    void set_first_name (std::string_view new_first)
    {
        std::scoped_lock _ (mutex);
        person_.set_first_name (new_first);
   }

    // Repeat for last_name

private:
    class __person;
    std::mutex mutex;
    mutable __person person_;
};

template<>
struct is_sync<person> : std::true_type {};
```

mutex

Implicit

locking

shared_mutex

```
class(mutex
get_first_name()
```

```
(std::string_view
```

{

```
};
```

std::string

void

```
last_name;
```

```
private:
```

```
default;
```

//

```
first_name,
```

first_name

{

Repeat

```
const
```

```
person()
```

```
set_first_name
```

```
public:
```

for

```
new_first;
```

```
first_name;
```

}

```
return
```

person

last_name

```
new_first)
```

$=$

$=$

)

```
class(shared_mutex
std::scoped_lock
```

`std::mutex`

set_first_name

```
is_sync<person>
```

```
person_.get_first_name();
person_.set_first_name
```

```
(std::string_view
get_first_name()
```

```
std::true_type
```

```
std::string
```

```
mutex;
```

[redacted]

void

{

```
};
```

```
template<>
```

last_name

```
(new_first);
```

struct

```
person()
```

```
person_;
```

```
(mutex);
```

```
default;
```

for

```
new_first)
```

=

mutable

```
public:
```

Repeat

class

class

{};

//

{

```
return
```

`const`

__person

```
__person;
```

person

```
private:
```

}

:

}

std::shared_lock

`std::unique_lock`

std::shared_mutex

locking

66

person

)

```
mutex;
```