



# Lightweight Data Race Detection

```
enum class check_type
{
    read,
    write
};

template<check_type type>
struct scoped_check
{
    scoped_check (check_state& check_state)
        : state (check_state)
    {
        if constexpr (type == check_type::read)
            read_started (state);
        else
            write_started (state);
    }

    ~scoped_check()
    {
        if constexpr (type == check_type::read)
            read_ended (state);
        else
            write_ended (state);
    }

    check_state& state;
};
```



# Wrapped Data Race Detection

```
struct(data_race_checker) person
{
    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;
};
```