



```
class person
public:
    std::string get_first_name() const
        auto sender = stdexec::then (stdexec::schedule (get scheduler()),
                                      [this] { return person.get first_name(); });
        auto [ret] = stdexec::sync wait (sender).value();
        return ret;
    void set_first_name (std::string new_first)
        auto sender = stdexec::then (stdexec::schedule (get scheduler()),
                                      [this, =] { return person.set first name (new first); });
        stdexec::sync wait (sender);
private:
    mutable __person person;
```









```
auto get_scheduler()
    static exec::static_thread_pool pool(1);
    return pool.get_scheduler();
```



```
auto get_scheduler()
{
    static exec::static_thread_pool pool(1);
    return pool.get_scheduler();
}
```

```
class person
public:
    std::string get_first_name() const
        auto sender = stdexec::then (stdexec::schedule (get_scheduler()),
                                     [this] { return person.get first name(); });
        auto [ret] = stdexec::sync_wait (sender).value();
       return ret;
   void set_first_name (std::string new_first)
        auto sender = stdexec::then (stdexec::schedule (get_scheduler()),
                                     [this, =] { return person.set_first_name (new_first); });
        stdexec::sync_wait (sender);
private:
   mutable __person person;
```





```
std::println ("\t\t\t\main tid: {}", std::this_thread::get_id());

person p;
std::println ("Name: {}", p.get_first_name());

std::thread t ([&]
{
    std::println ("\t\t\thread tid: {}", std::this_thread::get_id());

    p.set_first_name ("Dave");
    std::println ("Name: {}", p.get_first_name());
}
t.join();
```

```
main tid: 134711587358592
get tid: 134711584224832
Name:
thread tid: 126536174790208
set tid: 134711584224832
get tid: 134711584224832
Name: Dave
```