## - Exercise

In this lesson, we'll solve an exercise based on a variation of the sleep time of sleeper.

we'll cover the following ↑

• Problem statement

## Problem statement #

In the exercise, we need to vary the sleep time of the Sleeper class in the example from the previous lesson.

• Variations in the runtime are not synchronized undefined behavior.

```
#include <chrono>
#include <iostream>
#include <thread>
class Sleeper{
  public:
    Sleeper(int& i_):i{i_}{};
    void operator() (int k){
      for (unsigned int j = 0; j <= 5; ++j){
        std::this_thread::sleep_for(std::chrono::milliseconds(100));
        i += k;
      std::cout << std::this_thread::get_id() << std::endl;</pre>
  private:
    int& i;
int main(){
  std::cout << std::endl;</pre>
  int valSleeper= 1000;
  // Pass an argument here for sleep time variation
  std::thread t(Sleeper(valSleeper),5);
  // detach thread after each execution to run independently
```

```
t.join();
std::cout << "valSleeper = " << valSleeper << std::endl;

std::cout << std::endl;
}</pre>
```

In the next lesson, we'll discuss the solution to this exercise.