

Advanced Multithreading in C++ (WS 21/22)

Exercise 1

Please solve the following tasks by November 16, 2021. The results are not graded, but a solution is discussed on November 16, 2021.

Task 1

Implement a templated thread-safe stack. It should provide the following functionality:

- It stores the elements in an `std::stack` wrapped with `std::unique_ptr` internally.
- There is a method to push one element by copy semantics. This method only exists if the type is copy-able and copy-constructable without exceptions.
- There is a method to push one element by move semantics. This method only exists if the type is move-able and move-constructable without exceptions.
- There is a method to push one element that is already wrapped inside an `std::unique_ptr`.
- There is a method that takes an `std::array` of variable size (> 0) and pushes all elements. The elements must each be wrapped inside `std::unique_ptr`.
- There is a method that removes the upper most element and returns it within an `std::unique_ptr`.
- There is a method that takes as argument a reference to an object and populates it with the top element. This method only exists if the type is copy-assignable.
- There is a method that returns the number of currently stored elements any integral type.
- The thread-safe stack is move-constructable, but not assignable or copy-constructable.

Task 2

Consider the following control flows that each executes its operations in order. However, the different control flows can be arbitrarily interleaved. Find a total order of all operations that produces the results below. All variables are shared and initialized with 0.

1.1 $a = 5$	2.1 $b = 0$	3.1 $d = d * c$
1.2 $b = a + 3$	2.2 $a = a + 1$	3.2 $a = d \% 5$
1.3 $c = 12$	2.3 $a = a * 7$	3.3 $a = a + 1$
1.4 $d = a + b$	2.4 $d = c$	3.4 $b = a$

- $a = 42, b = 42, c = 12, d = 5$
- $a = 6, c = 9, c = 12, d = 15$
- $a = 1, b = 1, c = 12, d = 12$