

# Course Conclusion

Author's concluding words regarding the course.

The modern C++ was designed to be the better programming language for embedded and system programming.

As we have outlined, Modern C++ enables you to handle and address

- safety-critical systems.
- high-performance requirements combined with limited resources.
- work to be done in parallel.

For example, C++20 will likely get improved containers, which can be created at compile time. You can expect a `constexpr std::vector`, a `constexpr std::array`, and a `constexpr string`. But that is not all! Containers such as `std::flat_map` will overcome the weaknesses of a `std::map`.

The low latency study group (SG14) aims for the improvement of C++ in embedded and system programming. Experts from the game development, finance trading, and embedded domains have the same vision in mind and continue to develop C++ further.

If you want to peek into the future, you can read the proposals for future C++ versions here: [C++ Standards Committee Papers](#).

We hope this course might even help you participate in forming of the C++ future.