

# Where to Go from Here?

Want to learn more? There are more courses available to grow our C++ skill set.

## WE'LL COVER THE FOLLOWING



- Embedded Programming with Modern C++
- Generic Programming Templates in C++
- Modern C++ Concurrency: Get the most out of any machine
- C++ Standard Library including C++ 14 & C++ 17

## Embedded Programming with Modern C++ #

Embedded Programming with Modern C++ is highly valuable for each professional programmer.

In the past, embedded and system programming have had their pitfalls, but modern C++ has been designed to be a better language for this type of development, addressing the previous pitfalls/requirements explicitly. What are these requirements? Embedded systems deal with safety-critical systems, meaning they must guarantee high performance combined with limited resources, while also working in parallel.

The scope of this course goes beyond just embedded programming. Developers who write servers, games, or trading systems may especially benefit from this course because they also have to deal with safety-critical systems, high performance, reduced resources, and parallel execution in their jobs.

To get the most out of the course, you only need a basic understanding of C++. Building off this understanding, this course goes over all the essentials of embedded programming with Modern C++.

## Generic Programming Templates in C++ #

Templates are one of the many powerful features of Modern C++ programming. And with each new C++ standard, they become more important.

Templates provide an efficient way to make our code more flexible and reusable, this way we can avoid repeating code that would otherwise be identical except for different types. Templates also give us the ability to provide abstraction without a performance penalty.

However, templates can be difficult to apply, can be counterintuitive, and have tricky error messages.

This course gives us the necessary information to overcome these hurdles and dive into the advanced topics. We'll visit the basics such as function and class templates, and then explore the details to templates such as instantiation and fold expressions. Our journey also includes techniques based on templates, the design, and future directions of templates such as concepts. Based on this exhaustive tour of templates, a basic understanding of C++ is sufficient to master this course.

## Modern C++ Concurrency: Get the most out of any machine #

“Concurrency with Modern C++” is a journey through the present and upcoming concurrency features in C++.

- C++11 and C++14 have the basic building blocks for creating concurrent and parallel programs.
- With C++17 we have parallel algorithms from the Standard Template Library (STL). That means that most STL based algorithms can be executed sequentially, parallel, or vectorized.
- The concurrency story in C++ goes on. With C++20 we can hope for extended futures, co-routines, transactions, and more.

In addition to explaining the details of concurrency in modern C++, this course gives us many interactive code examples; therefore, we can combine theory with practice to get the most out of it.

Apart from theory, this course contains a lot of real-world scenarios and use-cases along with pitfalls and how to overcome them using best practices.

## C++ Standard Library including C++ 14 & C++ 17

#

C++ is one of the most prominent programming languages in the world. Along with optimized performance, the diverse functionality it offers has kept the language relevant for two decades. The C++ Standard Library, which is the collection of all the tools and utilities available in C++, received its latest update in 2017. Known as C++17, the update introduces many new features that further optimize our code.

Join us in this course as we venture into the vast world of C++17. We'll examine the fundamentals of the language, before moving on to more complex yet powerful tools. We can also experiment with the code provided and hence, gain a higher understanding of how things work. This course is perfect for anyone willing to fully grasp C++ and use it to its full potential. Welcome aboard!