

# What to Expect from the Course

A shorthand guide for what to expect from this course.

## WE'LL COVER THE FOLLOWING ^

- Highly Valuable for Any Professional Programmer \*
- Safety Critical Systems
- High Performance
- Reduced Resources
- Parallel execution

The typical audience for this course are embedded programmer, but any professional C++ programmer can benefit from the techniques and information presented in the course. Developers who write server, 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.

## Highly Valuable for Any Professional Programmer #

Modern C++ is designed to be the better programming language for embedded and system programming, so it addresses the requirements for embedded programming explicitly.

But, what is an embedded system? An **embedded system** is all from a tiny pacemaker to a large car or airplane. Unlike a general-purpose PC, Modern C++ typically serves one specific purpose and has no user-interface.

So, let's go over what Modern C++ can offer you as a programmer in this regard. Here are the main points:

Safety Critical Systems #

## Safety Critical Systems #

- Unified initialization
- Strongly typed enums
- User-defined literals

## High Performance #

- Type-traits
- constexpr
- Associative containers

## Reduced Resources #

- Move semantic
- Memory management
- Smart pointers

## Parallel execution #

- Memory Model
- Threading primitives such as Threads and Locks
- Tasks

---

As you can see, there are a lot of benefits to Modern C++. In the next lesson, we will go over the requirements for Embedded Programming.