

Running Source Code on your own machine

Teach yourself how to interact with and run source code examples of this course.

WE'LL COVER THE FOLLOWING ^

- Run the Programs

All source code examples are complete; that means, assuming you have a conforming compiler, you can compile and run them. The name of the source file is in the title of the listing. Only, if necessary, will I use the `using namespace std` directive in the source files.

Run the Programs

Compiling and running the examples is quite easy for the C++11 and C++14 examples in this course. Every modern C++ compiler should support them. For both the [GCC](#) and the [clang](#) compiler, the C++ standard must be specified as well as the threading library to link against. For example, the g++ compiler from GCC creates an executable program called `thread` with the following command-line: `g++ -std=c++14 -pthread thread.cpp -o thread`.

- `-std=c++14`: use the language standard C++14
- `-pthread`: add support for multithreading with the pthread library
- `thread.cpp`: source file
- `-o thread`: executable program

The same command-line works for the clang++ compiler. The Microsoft Visual Studio 17 C++ compiler supports C++14 as well. If you have no modern C++ compiler at your disposal, there are a lot of online compilers available. Arne Mertz' blog post [C++ Online Compiler](#) gives a great overview.

With C++17 and C++20, the story becomes quite complicated. I installed the [HPX \(High-Performance ParalleX\)](#) framework, which is a general purpose C++ runtime system for parallel and distributed applications of any scale. HPX has already implemented the [Parallel STL](#) of C++17 and many of the concurrency features of C++20. Please refer to the corresponding sections in the chapter [The Future: C++20](#), and read about how you can see the code examples in action.