C++ course (2020) outline

By Oleksiy Grechnyev, IT-JIM

Course level: Basic to intermediate, knowledge of any other programming language helps
[...] = Library/Technology of the day.

1. Introduction
   Why C++? Compilers vs interpreters. Hello World and other examples. C++ compilers (gcc, clang/llvm, CL) and IDEs. Boost. C++ bibliography. [Boost]
2. C++ Language Basics 1
   C++ versions (98, 03, 11, 14, 17). Built-in types. Operators. Statements. CMake example. using/typedef. auto. [OpenCV]
3. C++ Language Basics 2
   References. Pointers. const and constexpr modifier. Functions. Overloading. Headers. Namespaces. [Doxygen]
4. Classes 1
   Access modifiers : *public*, *private*, *protected*. Constructors. Destructors. Friends. Inheritance. Overriding, virtual and abstract methods. Multiple inheritance. [PlantUML]
5. Smart Pointers + Miscellanea 1
   Object life cycle. *unique_ptr*. *shared_ptr*. *weak_ptr*. Exceptions, try+catch. pair, any, optional. enum (class). Type casts. [Eigen]
6. Containers + Miscellanea 2
   array, vector, string, string_view, map. Iterators. Algorithms. Date + time. Random numbers. [RapidJSON]
7. Lambda expressions. IO streams.
   std::function. Function pointers. Functors. Lambda expressions. IO streams. [gtkmm]
8. Classes 2
   How to write your own class. Operator overloading. Copy + move constructors. [Profiling]
9. Using cmake and make
   CMake features. Static and dynamic libraries. Using make. [CUDA]
10. Concurrency.
    thread, async, future, atomic, mutex. [CTPL thread pool]
11. Templates. [TensorRT]
12. Move semantics.
    move, swap, rvalue references. [CBlas+Lapacke]