

ICS0025 Advanced C/C++

Overview

The primary goals of this course are:

1. Lead the students to a level of skills sufficient for participating in software projects basing on C++ versions 14 and higher.
2. Lead the students to a level of skills sufficient for implementing simpler Qt applications.

Number of ECTS credits awarded for this course is 6.0.

Presumptions

Generally, this course is for students who have successfully passed course ICS0017 “Fundamentals of C/C++”. If you have not, read the [syllabus](#) of this course and judge yourself are you prepared to participate.

Time and place

The course is offered in the spring term. The lessons (once per week, 4 * 45 minutes) include lectures as well as exercises and practical work.

According to the timetable, the lessons are on each Monday at 12:00 in room ICT-403. Exception: on March 18 the lesson starts at 16:00.

Language

The courseware and the lectures are in English.

Software support

In our computer class we use Windows 11 and Microsoft Visual Studio. The students are advised to download and install into their own laptop computers:

1. Visual Studio Community Edition (freeware, open link <https://visualstudio.microsoft.com/downloads/>). The newest version is from 2022.
2. Qt non-commercial open-source version (freeware, open link <https://www.qt.io/download>). The newest version is 6.2 but you may use also version 5.15.

Course themes

1. C++: some advanced features:
 - a. Class templates and function templates.
 - b. Additional types: *char32_t*, *__int32*, *auto*, etc.
 - c. *Typeid* and *decltype*.
 - d. Numerics library
 - e. Complex numbers.
 - f. *std::byte* type
 - g. class *any*.
 - h. Template *variant*.
 - i. Template *optional*.
 - j. Initializing and constructor initializer.
 - k. Pointers to functions and member functions.
 - l. Lambda expressions.

- m. Function wrappers.
 - n. Functors.
 - o. Copying and moving.
 - p. Smart pointers.
 - q. Rational numbers.
 - r. Standard classes for time handling.
 - s. Standard classes for random number generation.
2. STL containers:
- a. Vectors and iterating.
 - b. Lists.
 - c. Initializer lists.
 - d. Range-based *for*-loop.
 - e. Forward lists, queues, stacks, dequeues, priority queues.
 - f. Arrays and value arrays.
 - g. Pairs.
 - h. Tuples.
 - i. Maps.
 - j. Sets.
 - k. Multimaps and multisets.
 - l. Hashing and unordered maps.
 - m. Unordered sets, multisets and multimaps.
 - n. Bitsets
3. STL algorithms:
- a. Searching (*find*, *find_if*, *find_first_of*, *minmax*, *any_of*, etc.).
 - b. Algorithm *for_each*.
 - c. Filling (*generate*, *fill*, etc.)
 - d. Copying (*copy*, *copy_if*, etc.).
 - e. Replacing (*replace* and *replace_if*).
 - f. Removing (*remove* and *remove_if*).
 - g. Special (*unique*, *sort*, etc.)
4. Concurrency:
- a. Multithreading: organization and problems. Race conditions.
 - b. Launching, temporary stopping and ending of threads.
 - c. Exceptions in threads..
 - d. Mutexes and locks.
 - e. Calling a function just once.
 - f. Atomic variables.
 - g. Conditional variables.
 - h. Producer-consumer problem.
 - i. Futures.
 - j. Packaged tasks and promises.
 - k. Asynchronous input / output in Windows.
 - l. DLLs, explicit and implicit linking.
5. Qt:
- a. Installation
 - b. QtCreator and selecting of kits.
 - c. Creating of simple Qt widget applications with Qt designer.
 - d. Signals and slots.

- e. Events.
 - f. Qt threads.
 - g. Signals through thread boundaries.
 - h. Qt thread synchronization.
 - i. Qt strings and containers.
 - j. Qt input / output devices. File handling and remote device operations.
 - k. Setting parameters storing and retrieving.
 - l. Layouts.
 - m. Introduction to QML.
6. C++ version 20:
- a. Coroutines.
 - b. Ranges.
 - c. Modules.
 - d. Formatted output
 - e. Operator \Leftrightarrow .
 - f. Classes *jthread* and *stop_token*.
 - g. Synchronization tools: *latch*, *barrier*, *semaphore*

Courseware

The range of books on programming in C++ and Qt is wide and it is impossible to say which of them are the best to acquire the material of the current course. The books the instructor likes are as follows:

1. Marc Gregoire: Professional C++, Feb 2021, ISBN 978-1119695400 (very good for learning, there are also older editions from 2018, 2014 and 2011).
2. Marc Gregoire, Peter Van Weert: C++ Standard Library Quick Reference, Jul 2019, ISBN 978-1484249222 (very good for learning, there is also older edition from 2016).
3. Nicolai M. Josuttis: The C++ Standard Library: a Tutorial and Reference, Mar 2012, ISBN 978-0321623218 (good for learning).
4. Nicolai M. Josuttis: C++ 17 – the Complete Guide, Sep 2019, ISBN 978-3967300178 (very detailed).
5. Nicolai M. Josuttis: C++ 20 – the Complete Guide, Nov 2022, ISBN 978-3967309201 (very detailed).
6. Nicolai M. Josuttis, David Vandervoode: C++ Templates: the Complete Guide, Sep 2017, ISBN 978-0321714121 (not for beginners).
7. Anthony Williams: C++ Concurrency in Action, Jan 2019, ISBN 978-1617294693 (there is also older edition from 2012, good for learning).
8. Pavel Strakhov, Witold Wysota, Lorenz Haas: Game Programming Using Qt5. Apr 2018, ISBN 978-1788399999 (on the contrary to the title, the accent is not on the games but on the explaining of programming in Qt environment).

The full reference of C++ is on <http://www.cplusplus.com/reference/>. The instructor considers that <https://www.geeksforgeeks.org/c-plus-plus/> is the best on-line source for learning. For Qt reference open <https://doc.qt.io/qt-5/reference-overview.html>.

The participants will be supplied with course material including copies of [PowerPoint slides](#) and example code snippets. The complete printed notes, however, are not provided and the students are expected to participate the classwork.

Consultations

In case of problems write viktor.leppikson@liewenthal.ee

Assessment

To pass the course, the student has just to present all the three [courseworks](#) (i.e. three perfectly running applications) and demonstrate their work.