This is a basic course about programming in C++. The course is based on g++, make and Linux.

C++ is a language that makes it possible to combine high-level code with efficient execution on standard hardware. Contrary to common belief, the core of C++ is not object-oriented, but based on value semantics. C++ is a language in which one starts at low level, but which allows to move to higher levels of arbitrary abstraction. With a good programming style, low level can be abandonded quickly, and constructions of arbitrary complexity can be defined in a clean fashion.

Overview

- Structure of C++ programs. Separate compilation. Difference between compiling and linking. Use of make. Importance of good code layout. What should be written in comments.
- How to design good classes that hide their memory (and other resources) management. Different models for memory management (sharing, nonsharing). How to check for memory leaks. The different ways of referring to a variable (direct, through reference, through const reference, through rvalue reference, through pointer.)
- How to use the sequence containers (vector and list) in the STL. When to choose which container. Use of iterators and const_iterators. How to use the associative containers map and unordered_map. How to define your own containers using templates.
- Inheritance. When should one use it? When should one not use it? How should one use it? Construction and destruction with inheritance. Abstract classes.
- A bit of windows management and graphics (SFML, OpenGL). How to measure where a program spends its time.