AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 01/09/2026

Register

Jetbrains CLion Training

3 days (21 hours)

Overview

JetBrains CLion is a comprehensive IDE for developing in C/C++ with advanced assistance (navigation, refactoring, static analysis) and native CMake integration. This training course will help you speed up your development cycles, make your builds more reliable, and effectively debug modern projects, from embedded code to cross-platform applications.

The goal of our training is to make participants self-sufficient in configuring a C/C++ project in CLion, using productivity tools (search, inspections, refactorings), and mastering debugging and profiling. We cover common workflows around CMake, toolchains, and testing to standardize team practices.

The approach is decidedly practical: guided workshops, step-by-step demos, exercises on a common project (compilation, execution, testing, debugging). Deliverables include an operational CMake project, execution/debugging configurations, a checklist of best practices, and reproducible scripts/settings.

Objectives

- Install and configure CLion with a suitable toolchain.
- Create and structure a CMake project (targets, options, dependencies).
- Use navigation, inspections, and refactorings to improve code.
- Debug with breakpoints, watch, memory, and crash analysis.
- Set up tests and executions (GoogleTest/CTest) in the IDE.

Target audience

- C/C++ developers
- Software/embedded engineers
- Tech leads wishing to industrialize the CMake workflow

Prerequisites

- Basic knowledge of C or C++
- Understanding of compilation and linking
- Familiarity with the command line
- Basic knowledge of Git recommended

Technical requirements

- PC/Mac with at least 8 GB RAM (16 GB recommended)
- Windows, macOS, or Linux
- CLion installed (stable version)
- C/C++ toolchain: GCC/Clang or MSVC, and CMake
- Debugger: GDB/LLDB (depending on OS)

Jetbrains CLion training program

[Day 1 - Morning]

Getting started with CLion and configuring a C/C++ project

- Installation, activation, and overview of the interface (projects, editor, tools)
- Creating/importing a CMake project and understanding its structure
- Configuring toolchains (GCC/Clang/MSVC), CMake Profiles, and variables
- Configuring C/C++ standards and compilation options (warnings, optimization, debug)
- Hands-on workshop: Create a multi-file CMake project and compile it on your machine.

[Day 1 - Afternoon]

Productivity in the editor: navigation, refactoring, and inspections

- Quick navigation: symbols, files, call hierarchy, usages, and bookmarks
- Completion, templates, code generation, and contextual actions
- Essential refactorings: rename, extract, move, change signature, safe delete
- Inspections, quick fixes, formatting (ClangFormat), and style rules
- Hands-on workshop: Refactoring a module (names, function extraction, cleaning up warnings).

[Day 2 - Morning]

Advanced debugging with GDB/LLDB in CLion

- Configuring executions: Run/Debug, arguments, environment variables, working directory
- Breakpoints (conditional, log, exception), watchpoints, and thread management
- Memory inspection: variables, pointers, memory views, registers, and stack frames
- Debugging multi-process programs and attaching to a process
- Hands-on workshop: Diagnosing a crash (segfault) and fixing memory corruption.

[Day 2 - Afternoon]

Testing, quality, and static analysis

- Setting up and running tests (GoogleTest/Catch2) from the IDE
- Configuring CMake test targets and organizing test suites
- Coverage, targeted execution, diagnostics, and report reading
- Static analysis with clang-tidy and sanitizers (ASan/UBSan) via toolchain
- Hands-on workshop: Add a test suite and enable clang-tidy to fix defects.

[Day 3 - Morning]

CMake project management and dependencies

- Structuring a project: targets, include directories, compile definitions, and options
- Multi-config management: Debug/Release, presets, CMake cache, and clean generation
- Library integration: FetchContent, add_subdirectory, find_package
- Build tools: Ninja/Make, parallelization, logs, and linkage error resolution
- Hands-on workshop: Adding an external dependency and producing an executable target + a library.

[Day 3 - Afternoon]

Team workflow: Git, review, and automation in CLion

- Git in CLion: commit, branches, rebase/merge, conflict resolution
- Local code review: diff, history, annotations, and bug introduction search
- Shared run configurations, configuration files, and project best practices
- Productivity: Live Templates, macros, favorite actions, and environment configuration
- Hands-on workshop: Setting up a Git workflow (feature branch, rebase, conflict resolution) on a mini-project.

Target companies

This training is intended for both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology or to acquire specific business knowledge or modern methods.

Placement at the start of training

The placement test at the start of the training course complies with Qualiopi quality criteria. Upon final registration, learners receive a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training course, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and discussion sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training course.