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

Mis à jour le 28/07/2023

S'inscrire

Formation C++ Moderne

3 jours (21 heures)

Présentation

Cette formation C++ moderne s'adresse aux personnes qui maitrisent déjà les bases du langage C++ et souhaitent se perfectionner pour gagner en robustesse, performante et productivité.

L'écosystème C++ évolue en effet de façon intense depuis 2011, au rythme d'une nouvelle norme tous les 3 ans. Côté hardware, la multiplication des cœurs ces dernières années implique aussi de repenser sa manière de programmer. Tout cela pose un vrai défi d'adaptation, tant pour assimiler les nouveautés que pour se défaire de pratiques désormais obsolètes.

C'est pour répondre à ce défi que cette formation a été conçue. En mettant en lumière les principes fondamentaux qui guident l'évolution du langage, il devient beaucoup plus facile de progresser sans se perdre dans la complexité. Et donc d'écrire un code plus simple et plus fiable à la fois.

Comme pour toutes nos formations, celle-ci vous présentera la toute dernière version de C++, à savoir C++ 20.

Objectifs

- Assimiler et mettre en œuvre les principes du C++ moderne
- Se familiariser avec les évolutions de C++ jusqu'à C++20
- Maitriser les concepts avancés de programmation parallèle et concurrente

Public visé

- Développeurs
- Architectes

Pré-requis

- Connaissance basique en C++ (notion d’objet et pointeur)

Programme de la formation C++ Moderne et Multithreading

Révision des bases du langage

- Définition du langage C++
- Les fonctions: inline, static, const, constexpr, virtual, consteval, ...
- Plain Old Data (POD) : struct vs class
- Déclaration vs définition (One Definition Rule)
- La STL : ses conteneurs et ses algorithmes
- Principes de base des templates, différence avec constexpr
- Comportements indéfinis (Undefined behavior)

Fondements du C++ moderne

- Présentation des C++ Core Guidelines
- Évolutions dans la syntaxe (C++14, C++17, C++20)
- Gestion robuste des ressources (RAII, smart pointers, scope guards)
- Notion de propriété (ownership), sémantique de déplacement (std::move)
- Sémantique de valeur / entité
- Simplifier les initialisations complexes avec des lambdas
- Un code plus robuste et expressif avec des alias fortement typés (strong typedefs)

Programmation orientée objet avancée

- Polymorphisme statique et dynamique
- Principe de substitution de Liskov
- Héritage vs composition
- Classes abstraites, PIMPL
- POO sans héritage
- Principes SOLID

Notions de programmation fonctionnelle

- Les lambdas
- Immutabilité (const) et fonctions pures
- Type optionnel avec std::optional
- Type composite avec std::variant
- Type erasure : std::string_view, std::span

Multithreading et scalabilité

- Parallélisme vs concurrence
- Race condition, impact du mot-clé const
- Protection des sections critiques (std::mutex, std::condition_variable, ...)
- Enjeux de la scalabilité (loi d'Amdahl)
- Algorithmes parallèles de la STL (C++17)
- Programmation lock free (std::atomic)
- Programmation asynchrone : promise, future, coroutines
- Map, filter, reduce (Qt Concurrent)
- Thread Local Storage (thread_local)
- Détecter les problèmes (outils disponibles)
- Autres approches : OpenMP, CUDA, Boost.Interprocess, gRPC

Outils modernes du développeur C++

- Bien configurer son compilateur
- Tests unitaires avec Catch2
- Utilisation de CMake et des CMake presets
- Formatage automatique du code avec clang-format
- Gestion de dépendances avec Conan
- Couverture de code avec gcov / lcov
- Analyseurs statiques
- Compiler explorer (Godbolt)

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l’acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.