

— Public	Développeurs C++ souhaitant connaître les nouveautés définies par le standard C++11.
— Durée	3 jours - 21 heures
— Pré-requis	Bonnes connaissances du langage C++. Une expérience pratique de la programmation avec ce langage est recommandée.
— Objectifs	Appréhender les nouveautés et les améliorations du langage C++ Utiliser les lambda expressions Améliorer ses performances avec la sémantique de déplacement Maîtriser les allocations-destructions d'objets avec les smart pointers Développer une application multithreadée
— Méthodes pédagogiques	Pour bien préparer la formation, le stagiaire remplit une évaluation de positionnement et fixe ses objectifs à travers un questionnaire. La formation est délivrée en présentiel ou distanciel (e-learning, classe virtuelle, présentiel et à distance). Le formateur alterne entre méthodes démonstratives, interrogatives et actives (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification. Cette formation est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par Audit Conseil Formation.
— Moyens techniques	1 poste de travail complet par personne De nombreux exercices d'application Mise en place d'ateliers pratiques Remise d'un support de cours Passage de certification(s) dans le cadre du CPF Remise d'une attestation de stage
— Modalité d'évaluation des acquis	Evaluation des besoins et objectifs en pré et post formation Evaluation technique des connaissances en pré et post formation Evaluation générale du stage
— Planning	Du 08/09/2025 au 10/09/2025 Du 10/11/2025 au 12/11/2025
— Délai d'accès	L'inscription à cette formation est possible jusqu'à 5 jours ouvrés avant le début de la session
— Accessibilité handicapés	Au centre d'affaires ELITE partenaire d'ACF à 20 m. Guide d'accessibilité à l'accueil.

L'AVÈNEMENT DE C++11

- Les différentes normes C++98, C++03, C++11, C++14, C++17, C++20.
- Les nouveautés de C++11 et les objectifs de cette norme. Le devenir de Boost, STL.
- La question de la compatibilité des codes anciens.
- La disponibilité des outils de développement (compilateurs, débogueurs, IDE...).

LES AMÉLIORATIONS DU LANGAGE

- Les enum fortement typées.
- Le mot-clé auto et decltype pour simplifier le typage.
- La boucle basée sur un intervalle.
- Constexpr pour une évaluation à la compilation.
- Templates variadiques, concepts de C++20 soucoupe...
- Coroutines de C++20.
- Les modules de C++20.

LES MODIFICATIONS AU NIVEAU DES CLASSES

- La délégation de constructeurs, les contraintes liées à l'héritage.
- La nouvelle sémantique du déplacement et le constructeur par déplacement (move constructor).
- Les directives override, final =delete, =default.
- Notion de module C++20

LA PROGRAMMATION FONCTIONNELLE AVEC LES EXPRESSIONS LAMBDA

- Déclaration, typage, implémentation et utilisation.
- L'intérêt d'auto avec les lambda-expressions.
- La gestion des fermetures (closures), avec capture par valeur ou par référence des variables liées au contexte.

L'UTILISATION DES THREADS

- Déclaration et exécution d'un thread. Attente de fin d'exécution avec `join()`.
- La gestion des données locales à un thread.
- Récupérer un résultat avec `future<>` et `async()`.
- Choisir parmi les différents verrous de la STL.

AUTRES NOUVEAUTÉS DE LA BIBLIOTHÈQUE STANDARD

- Ranges avec `std::view` pour évaluations à la volée de C++20.
- La gestion du temps avec le namespace `chrono`.
- Le nouveau conteneur `tuple`.
- Conteneurs `unordered_set`, `unordered_map` à base de hachage.
- Formatage de `string` avec C++20.

LA GESTION MÉMOIRE ET LES CONTENEURS

- Les smart pointers : `shared_ptr`, `weak_ptr`, `unique_ptr`. Usage conjoint avec la STL.

NOUS CONTACTER

Siège social

16, ALLÉE FRANÇOIS VILLON
38130 ÉCHIROLLES

Téléphone

04 76 23 20 50 - 06 81 73 19 35

Centre de formation

87, RUE GÉNÉRAL MANGIN
38000 GRENOBLE

E-mail

contact@audit-conseil-formation.com

Suivez-nous sur les réseaux sociaux, rejoignez la communauté !



ACF Audit Conseil Formation



@ACF_Formation



ACFauditconseilformation