Formation Développement C++ avancé

Approfondissez vos connaissances en programmation C++

N° ACTIVITÉ : 11 92 18558 92 | TÉLÉPHONE : 01 85 77 07 07 | E-MAIL : inscription@hubformation.com

De nombreux experts s'accordent à reconnaitre que la maitrise de C++ suppose une longue pratique et une forte implication dans des projets complexes. Les normes récentes (C++ 11/14/17) ont d'ailleurs sensiblement amélioré la puissance d'expression et les possibilités de conception qu'offre ce langage. La finalité de cette formation intensive est de raccourcir de façon significative la durée de cet apprentissage pour tirer parti des nouveautés du langage et ainsi développer des applications encore plus puissantes.

Référence	LNG201
Durée	4 jours (28h)
Tarif	2 590 €HT

Objectifs

| Maîtriser les mécanismes avancés de C++ (héritage multiple / virtuel, programmation générique, méta-programmation, ...)
| Appréhender et mettre en oeuvre les techniques de la nouvelle norme C++11/14 telles que les lambda-expressions, la programmation fonctionnelle ou la notion de move semantics
| Savoir pratiquer l'approche TDD (Test Driven Development) en C++
| Découvrir et maîtriser la programmation parallèle et synchronisée introduite par C++11

Public

| Développeurs C++
| Chargés de développement d'applications informatiques

Prérequis

| Avoir suivi la formation Les fondamentaux du développement C++ (LA200) ou maîtriser le langage de programmation orientée objet C++

Programme de la formation

Nouveautés fondamentales de la norme C++11/14
| nullptr_t et le littéral nullptr
| Les nouveaux types, littéraux et séparateurs
| Variables templates
| Initialisation uniforme
| Initialisation de tableaux et collections
| Parcours unifié des tableaux et conteneurs
| Listes d'initialisation avec initializer_list
| La boucle range based for
| Énumérations fortement typées
| Types normalisés et variantes (uint_8, uint64_t, ...)
| Contrôle de l'alignement mémoire
| Inférence de types et de signatures avec auto
| C++14 et déduction étendue
| Les nouveaux spécificateurs de classe (override, default, delete, final)
| Constructeur délégué
| Constructeur hérité
| Alias et using
| Expressions constantes avec constexpr

SESSIONS PROGRAMMÉES

A DISTANCE (FRA)
- du 15 au 18 juillet 2025
- du 20 au 23 octobre 2025

PARIS
- du 15 au 18 juillet 2025
- du 20 au 23 octobre 2025

AIX-EN-PROVENCE
- du 15 au 18 juillet 2025

BORDEAUX
- du 20 au 23 octobre 2025

GRENOBLE
- du 20 au 23 octobre 2025

LILLE
- du 20 au 23 octobre 2025

LYON
- du 20 au 23 octobre 2025

NANTES
- du 20 au 23 octobre 2025

RENNES
- du 20 au 23 octobre 2025

ROUEN
- du 20 au 23 octobre 2025

SOPHIA-ANTIPOLIS
- du 15 au 18 juillet 2025

VOIR TOUTES LES DATES

| Gestion du temps, l'espace de nom chrono

Move semantics
| Copie versus déplacement
| Value et RValue reference
| La fonction move
| Move constructor et move assignement operator
| Héritage et move constructor / op=
| STL C++11 et swap / move
| Signature reference qualifiers
| Mauvaises pratiques

Gestion des ressources
| Resource Acquisition Is Initialization (RAII)
| L'opérateur -> avec ou sans généricité
| Propriété et transfert de responsabilité
| La classe unique_ptr
| Unique_ptr et tableaux dynamiques
| Comptage de références avec la classe shared_ptr
| Custom deleter
| Les fonctions make_unique (C++14) et make_shared
| La classe weak_ptr et le référencement circulaire

Gestion avancée des exceptions
| Principes
| Dynamique
| Traiter une exception
| Concevoir et hiérarchiser les exceptions
| Traitement par défaut
| Les exceptions prédéfinies
| Abraham's Exception safety guarantees
| La clause C++11 noexcept

Héritage avancé
| Surcharge et ambiguïtés
| Héritage public et redéfinition privée
| Héritage privé et protégé
| Héritage multiple
| Héritage en diamant
| Héritage virtuel et dominance

Programmation fonctionnelle avec C++ 11/14
| Problématique de l'abonnement
| Les classe function et mem_fn
| Binding, placeholders
| Adaptateurs de références
| Les lambda-expression

Utilisation avancée de la généricité
| Typage multiple
| Inférence des retours avec decltype
| Paramétrage / spécialisation des méthodes
| Perfect forwarding avec std::forward
| Héritage / containment et généricité
| Méta-programmation
| L'idiome CRTP Curiously Recursive Template Pattern
| Typologie C++ et classes de traits
| Assertions statiques avec static_assert
| Extended friend declaration
| Les variadic templates, pattern matching et héritage multiple
| SFINAE Substitution Failure Is Not An Error

Nouveautés C++11 de la librairie standard
| Les nouveaux itérateurs cbegin
| Les tableaux à taille fixe avec std ::array
| Les nouvelles collections associatives : unordered_map/set

HUB Formation - 14 avenue de l'Opéra - 75001 Paris
01 85 77 07 07 - www.hubformation.com - SIREN 525 287 785

| Singly-Linked Lists
| Le conteneur tuple
| Adaptateurs d'itérateurs, stream itérateurs
| Les nouveaux algorithmes ensemblistes

Multithreading
| Principes
| Démarrage et détachement d'un thread
| La classe std ::call_once
| L'espace de noms this_thread
| Futures / promises et packaged_task
| Programmation asynchrone avec async
| Politiques de démarrage
| Comparaison thread versus future
| Partage de ressources et mécanismes de synchronisation
| Mutexes et données atomiques
| Unique-lock et lock_guard

Approche Test Driven Design (TDD) en C++ avec GoogleTest
| Concepts
| Assertions et tests élémentaires
| Tests multiples : suites de tests et Test Cases
| Préparation et mutualisation des tests avec les Test fixtures

Méthode pédagogique

Chaque participant travaille sur un poste informatique qui lui est dédié. Un support de cours lui est remis soit en début soit en fin de cours. La théorie est complétée par des cas pratiques ou exercices corrigés et discutés avec le formateur. Le formateur projette une présentation pour animer la formation et reste disponible pour répondre à toutes les questions.

Méthode d'évaluation

Tout au long de la formation, les exercices et mises en situation permettent de valider et contrôler les acquis du stagiaire. En fin de formation, le stagiaire complète un QCM d'auto-évaluation.

Suivre cette formation à distance

Voici les prérequis techniques pour pouvoir suivre le cours à distance :

| Un ordinateur avec webcam, micro, haut-parleur et un navigateur (de préférence Chrome ou Firefox). Un casque n'est pas nécessaire suivant l'environnement.
| Une connexion Internet de type ADSL ou supérieure. Attention, une connexion Internet ne permettant pas, par exemple, de recevoir la télévision par Internet, ne sera pas suffisante, cela engendrera des déconnexions intempestives du stagiaire et dérangera toute la classe.
| Privilégier une connexion filaire plutôt que le Wifi.
| Avoir accès au poste depuis lequel vous suivrez le cours à distance au moins 2 jours avant la formation pour effectuer les tests de connexion préalables.
| Votre numéro de téléphone portable (pour l'envoi du mot de passe d'accès aux supports de cours et pour une messagerie instantanée autre que celle intégrée à la classe virtuelle).
| Selon la formation, une configuration spécifique de votre machine peut être attendue, merci de nous contacter.
| Pour les formations incluant le passage d'une certification la dernière journée, un voucher vous est fourni pour passer l'examen en ligne.
| Pour les formations logiciel (Adobe, Microsoft Office...), il est nécessaire d'avoir le logiciel installé sur votre machine, nous ne fournissons pas de licence ou de version test.
| Horaires identiques au présentiel.

Accessibilité

Les sessions de formation se déroulent sur des sites différents selon les villes ou les dates, merci de nous contacter pour vérifier l'accessibilité aux personnes à mobilité réduite.
Pour tout besoin spécifique (vue, audition...), veuillez nous contacter au 01 85 77 07 07.