

— Public	Concepteurs et développeurs d'applications en C++, chefs de projets, architectes logiciels.
— Durée	4 jours - 28 heures
— Pré-requis	Bonnes connaissances en développement C++. Expérience indispensable.
— Objectifs	Découvrir les nouveautés apportées par les versions Maîtriser la gestion de la mémoire, des pointeurs et des références Implémenter la généricité en C++ Découvrir la bibliothèque standard STL Utiliser les apports de la norme C++11
— 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 11/09/2025 Du 03/11/2025 au 06/11/2025 Du 08/12/2025 au 11/12/2025
— 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.

RAPPELS

- Classes d'allocation mémoire.
- Construction, initialisation, embarquement d'objets.
- Les fuites mémoire.
- Constance, le mot-clé mutable, Lazy Computation.
- Amitié (friendship) C++ et contrôle d'accès.
- Destruction virtuelle.
- Stratégie de gestion des exceptions.
- Les espaces de nommage (namespace).

LES NOUVEAUTÉS LANGAGE DE C++11

- nullptr et autres littéraux.
- Les directives =delete, =default.
- Délégation de constructeurs.
- Les énumérations "type safe".
- Le mot-clé auto et boucle sur un intervalle.
- Référence rvalue et impact sur la forme normale des classes C++.
- Les lambda expressions.

GESTION DES OPÉRATEURS

- Opérateurs binaires et unaires.
- L'opérateur d'indirection, cas d'usage.
- L'opérateur de référencement.
- Les opérateurs d'incrément/décément préfixés et post-fixés.
- Les autres opérateurs : comparaison, affectation...
- La surcharge de l'opérateur [], des opérateurs d'insertion (<<) et d'extraction (>>).
- Les foncteurs et la surcharge de l'opérateur (), avantage par rapport aux fonctions.

CONVERSION ET RTTI

- Opérateurs de conversion. Constructions implicites, le mot-clé explicit.
- Les opérateurs de casting `const_cast`, `static_cast`, `reinterpret_cast`.
- Conversion dynamique et Runtime Type Information.
- L'opérateur `typeid`, les exceptions liées.
- La classe `type_info`.
- Contrôle du "downcasting" à l'aide de l'opérateur `dynamic_cast`.

LA GÉNÉRICITÉ

- Introduction aux patrons de classe. Généricité et préprocesseur.
- Fonction générique. Classe générique. Composition générique. Généralisation générique.
- Spécialisation partielle et totale.
- Introduction à la méta-programmation.
- La généricité, principe fédérateur des bibliothèques STL et Boost.

LA STL (STANDARD TEMPLATE LIBRARY)

- Composants de la STL : types complémentaires, conteneurs, algorithmes, itérateurs, objets fonctions, les adaptateurs.
- Les chaînes de caractères STL, la classe `template basic_string` et ses spécialisations.
- Les conteneurs séquentiels et associatifs : définition, rôle et critères de choix.
- Les allocateurs et la gestion de la mémoire des conteneurs.
- Les méthodes d'insertion, de suppression, d'itération et d'accès aux principaux conteneurs : `Vector`, `List`, `Set`, `Stack`...
- Le concept d'itérateur. Parcours d'un conteneur.
- Les différents groupes d'algorithmes STL : non mutables, mutables, de tri et de fusion, numériques.
- Manipulation de conteneurs (manipulation, recherche de valeurs...).
- Paramétrer les algorithmes génériques par des objets "fonction".
- Les "adaptateurs" et la modification du comportement d'un composant.
- La STL et les traitements sur les flux (fichiers, mémoire...).
- Principe du RAII : les pointeurs automatiques et la classe `auto_ptr`.
- Les exceptions standard de la STL.

LES NOUVEAUTÉS C++11 DE LA BIBLIOTHÈQUE STANDARD

- Evolution historique : Boost --> TR1 --> C++11.
- Les nouveaux conteneurs : `array`, `forward_list`, `unordered_set`, `unordered_map`.
- La classe `tuple`.
- Les pointeurs intelligents (smart pointer) : `shared_ptr`, `weak_ptr`, `unique_ptr`.
- Les nouveaux foncteurs et binders.
- Introduction à la gestion des threads.
- Les expressions régulières.

BOOST ET SES PRINCIPES

- La Pointer Container Library (destruction des données pointées d'un conteneur).
- Les structures de données `boost::any` et `boost::variant`.
- Programmation événementielle (connexions et signaux).
- Gestion des processus, mécanismes de communication interprocessus et mémoire partagée.

UTILISATION AVANCÉE DE L'HÉRITAGE

- Héritage versus embarquement. Héritage privé. Héritage protégé.
- Exportation de membres cachés avec la Clause Using.
- Héritage multiple et gestion des collisions de membres.

- Héritage en diamant. Héritage virtuel et dynamic_cast.
- Principes de conception : substitution de Liskov, principe d'ouverture/fermeture, inversion des dépendances.
- Règles d'implémentation des interfaces en C++.

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