retengr

C++ avancé

Durée : 4.5 jours

Date de mise à jour : 12/05/2026

Méthode pédagogique

La formation se compose de travaux pratiques (mises en situation, débats, exercices).

Une évaluation quotidienne de l'acquisition des connaissances de la veille est effectuée.

Une synthèse est proposée en fin de formation.

Une évaluation à chaud sera proposée au stagiaire à la fin du cours.

Un support de cours (version électronique) sera remis à chaque participant comprenant les slides sur la théorie, les exercices.

Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

Présentation

Cette formation approfondit vos connaissances en C++. Vous maîtriserez les fonctionnalités avancées du langage, notamment les apports de C++11, la programmation générique et les techniques de gestion des opérateurs.

40 rue des Filatiers, 31000 Toulouse - Organisme de formation : 73 31 08023 31 - SIRET : 812 673 903 00017

Objectifs

- Maîtriser les nouveautés du C++11
- Gérer les opérateurs et les conversions avancées
- Mettre en œuvre la programmation générique (templates)
- Utiliser le RTTI et les techniques avancées du C++

Audience

Développeurs

Le formateur

Le formateur est un expert du domaine qui intervient sur le sujet depuis plusieurs années en formation mais aussi en conseil.

Pré-requis

Maîtrise des bases du C++ (classes, héritage, polymorphisme)

Expérience pratique en développement C++

Programme

Rappels [3.5h]

- Classes d'allocation mémoire.

40 rue des Filatiers, 31000 Toulouse - Organisme de formation : 73 31 08023 31 - SIRET : 812 673 903 00017

- 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 [3.5h]

- 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.
- Travaux pratiques
- Réécriture d'un code C++ existant en C++11, comparaison des deux implémentations.

Gestion des opérateurs [3.5h]

- Opérateurs binaires et unaires.
- L'opérateur d'indirection, cas d'usage.
- L'opérateur de référencement.
- Les opérateurs d'incrémentation/décrémentation 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.
- Travaux pratiques
- Création de foncteurs et de proxies (libération mémoire, comptage de références) avec les opérateurs étudiés.

40 rue des Filatiers, 31000 Toulouse - Organisme de formation : 73 31 08023 31 - SIRET : 812 673 903 00017

Conversion et RTTI [3.5h]

- 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.
- Travaux pratiques
- Mise en œuvre des idiomes "is-a" et "is-kind-of" avec dynamic_cast.

La généricité [3.5h]

- 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 librairies STL et Boost.
- Travaux pratiques
- Démarrage de l'étude de cas qui sera complétée avec la STL et Boost. Mise en œuvre de la composition et de la généralisation génériques. Création de plug-ins génériques.

La STL (Standard Template Library) [3.5h]

- 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.

40 rue des Filatiers, 31000 Toulouse - Organisme de formation : 73 31 08023 31 - SIRET : 812 673 903 00017

- Les différents groupes d'algorithmes STL : non mutants, mutants, de tri et de fusion, numériques.
- Manipulation de conteneurs (manipulation, recherche de valeurs...).
- Paramétrer les algorithmes génériques par des objets "fonction".
- Les "adapteurs" 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.
- Travaux pratiques
- Implémentation des relations avec les collections de la STL. Utilisation d'algorithmes standard quelconques.

Les nouveautés C++11 de la librairie standard [3.5h]

- 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.
- Travaux pratiques
- Mise en œuvre de la robustesse avec les smart pointers. Utilisation d'expressions régulières.

Boost [3.5h]

- 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.
- Travaux pratiques
- Amélioration de l'implémentation de l'étude de cas par l'utilisation la Pointer Container Library.

Utilisation avancée de l'héritage [3.5h]

- 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++.
- Travaux pratiques
- Combinaison de l'héritage multiple, privé et de l'exportation pour concevoir des classes robustes et hautement évolutives.

Modalités et délais d'accès à la formation

Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation, en interentreprises, dans la limite des places disponibles. Pour les formations organisées en intra entreprise, la liste des participants peut être modifiée jusqu'à 24h ouvrés avant le début de la formation.

Accessibilité

RETENGR facilite l'accessibilité de ses formations.

Cette formation est accessible aux personnes en situation de handicap.

Si vous avez un besoin d'accès spécifique, contactez Céline BOURREIL (celine.bourreil@retengr.com) qui étudiera avec Handifiel's (notre référent handicap) votre demande et vous proposera les meilleures solutions

Vous allez nous adorer si comme nous vous pensez que...

Une formation doit être au service de la performance du collaborateur et de l'entreprise

Ceci nécessite une quête constante d'excellence de la part de l'organisme formateur avec une adaptation systématique aux enjeux de l'entreprise, la mise à jour régulière des supports de cours et une veille technologique indispensables pour toujours être à la pointe du domaine.

L'expertise technique est aussi importante que les qualités pédagogiques

Nos formateurs sont tous des experts de leur domaine. Mais qu'ont-ils de plus que les autres ? Nous les sélectionnons en plus pour leurs qualités de pédagogue et leurs méthodes d'enseignements. Nous plaçons les qualités pédagogiques au même niveau que l'expertise afin que nos stagiaires tirent le meilleur de leurs formations.

L’excellence naît de l’excellence

Beaucoup de nos clients se classent parmi les leaders de leurs industries respectives ou parmi les start-ups les plus prometteuses. Nous savons que former les collaborateurs de telles entreprises nécessite de prêter attention à chaque détail en prodiguant un accompagnement à la hauteur de l’ambition de nos stagiaires. C’est pourquoi nous savons faire des leaders d’aujourd’hui les champions de demain !

retengr

Faire du leader d'aujourd'hui, le champion de demain