

C++ NIVEAU 2

Durée	4 jours	Référence Formation	4-LC-CPAV
--------------	----------------	----------------------------	------------------

Objectifs

Grace à cette formation vous pourrez
Consolider vos connaissances dans le développement C++
Disposer de connaissance pour rendre efficaces vos développements

Participants

Concepteurs et développeurs d'applications en C++, chefs de projets, architectes logiciels.

Pré-requis

Il est nécessaire de Disposer de bonnes connaissances en développement C++.

Moyens pédagogiques

Réflexion de groupe et apports théoriques du formateur
Travail d'échange avec les participants sous forme de réunion-discussion
Utilisation de cas concrets issus de l'expérience professionnelle
Validation des acquis par des questionnaires, des tests d'évaluation, des mises en situation et des jeux pédagogiques
Remise d'un support de cours.

PROGRAMME

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...

CAP ÉLAN FORMATION

www.capelanformation.fr - Tél : 04.86.01.20.50

Mail : contact@capelanformation.fr

Organisme enregistré sous le N° 76 34 0908834

version 2024

- 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 mutants, mutants, 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 LIBRAIRIE 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++.



CAP ÉLAN FORMATION

www.capelanformation.fr - Tél : 04.86.01.20.50

Mail : contact@capelanformation.fr

Organisme enregistré sous le N° 76 34 0908834
version 2024