

C++ avancé

Améliorer et industrialiser le processus de développement d'applications C++

Cette formation couvre tous les aspects avancés de C++ nécessaires pour la réalisation d'applications.

Détails

- Code : PR-CPPA
- Durée : 4 jours (28 heures)

- Public
- Chefs de projets
 - Développeurs

- Pré-requis
- Programmer en objets avec C++

Objectifs

- Approfondir la connaissance du langage

Programme

Introduction sur les pièges C++

- Présentation des principaux pièges : pointeurs, MLK, compteurs de référence, constructeurs, virtualités, etc.
- Expérience de grand projets
- Présentation des pièges classiques : allocation de données globales, directives de pré-compilation, croisement des fichiers entêtes
- Pas d'atelier dans cette présentation

Gestion de la mémoire avancée de C++

- Allocation dynamique de la mémoire
- Libération de la mémoire
- Pièges « courant » liés à l'utilisation de la mémoire en C++
- Atelier : études de cas sur la gestion de la mémoire

Gestion des pointeurs mémoire

- Pointeurs et pointeurs de pointeurs
- Pièges « courants » liés à l'utilisation des pointeurs
- Ecrire un meilleur code en utilisant les pointeurs
- Atelier : étude de cas sur les pointeurs

Les classes

- Constructeur et destructeur
- Les constructeurs et la virtualité
- Le destructeur virtuel et l'héritage
- Constructeur par copie : utilité et règle
- Atelier : constructeur par copie, destructeur virtuel etc.

Gestion des exceptions

- Principe de la gestion des exceptions

- Mise en oeuvre
- Dangers liés à la gestion des exceptions
- Atelier : études de cas

Les classes, mécanismes avancés

- Mécanismes avancés
- Héritage privé et multiple, l'héritage virtuel
- La classe abstraite
- Principes des interfaces en P.O.O.
- Simuler des interfaces avec des classes
- Problèmes constatés dans les projets dans certains cas avec les classes
- Atelier : études de cas

La gestion des Smart Pointer, piège

- Principe des « Smart Pointers »
- Controverse
- Opérateurs de conversion de types
- Atelier : les « Smart Pointers »

Les outils de test d'un programme C++

- Introduction aux tests des processus
- Debugging avancé
- Détection de fuites mémoire
- Détection des violations d'accès à la mémoire
- Effets de bord
- Atelier : tester un code C++ pour le rendre efficace

Les règles C++

- Pourquoi des règles de programmation
- Liste des règles

Modalité

- Stage pratique en présentiel
- Stage pratique en distanciel
- Nombre de stagiaires minimum : 4
- Nombre de stagiaires maximum : 10

Méthodes pédagogiques

- Exposés
- Cas pratiques

Profils des intervenants

- Toutes nos formations sont animées par des consultants-formateurs expérimentés et reconnus par leurs pairs.

Modalités d'évaluation

- Evaluation des acquis de la formation par le biais de cas pratiques et/ou mises en situation.
- Attestation de formation remise à chaque participant.

Démarche qualité

- Questionnaire d'évaluation de satisfaction à chaud complété par chaque participant à l'issue de la formation.

Moyens pédagogiques

- Salle équipée de PC (1 poste par stagiaire), vidéo-projecteur.
- Espace de pause.

Dernière mise à jour le 14/12/2021