

# Programmation C++, perfectionnement

Réaliser des applications efficaces

## L'avis de l'expert

Le C++ est un langage très technique mettant à rude épreuve les nerfs des développeurs d'applications. La syntaxe du C, conjuguée avec la mécanique des pointeurs mémoires, la programmation orientée objet et les classes, demandent une attention de tous les instants dans l'utilisation du C++. Une grande rigueur dans l'approche de ces développements et une bonne connaissance des problèmes « mécaniques » liés à l'utilisation du C++ sont nécessaires. Cette formation a été bâtie suivant une approche « études de cas » afin de faire profiter des expériences acquises dans le développement d'applications importantes.

## Public

Développeurs d'applications.

## Pré-requis

Plusieurs mois de pratique du langage C++.

## Travaux pratiques

Le cours se déroulera sur des stations de travail sous Windows/Visual C++.

## Itinéraire pédagogique

### Introduction sur les pièges C++

- Présentation des principaux pièges : pointeurs, MLK, compteurs de référence, constructeurs, virtualités, etc.
- Expérience de grands projets.
- Présentation des pièges classiques : allocation de données globales, directives de pré-compilation, croisement des fichiers en-têtes.

### Gestion de la mémoire avancée de C++

- Allocation dynamique de la mémoire.
- Libération de la mémoire.
- Fuites mémoire (memory leaks).
- Débordement de tableaux et pagination mémoire des systèmes.
- Pièges « courants » liés à l'utilisation de la mémoire en C++.

#### Atelier

Etudes 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.
- Règles d'utilisation des pointeurs et de leur initialisation.

#### Atelier

Etude de cas sur les pointeurs.

POP

1 880€HT



compris

Dates

Format 4 Jours

Paris - La Défense  
6/03/2007, 29/05/2007,  
3/07/2007, 18/09/2007,  
27/11/2007

Ce stage peut-être réalisé en  
intra entreprise.  
Contactez-nous  
au 01 42 37 12 34

### Les classes

- Constructeur et destructeur.
- Utilisation de la liste d'initialisation.
- Constructeur de copie et surcharge de l'opérateur « = », règles.
- Les constructeurs et la virtualité.
- Le destructeur virtuel et l'héritage.

#### 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

*Travaux sur l'implémentation d'une gestion des exceptions.*

---

### 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

*Etudes de cas.*

---

### La gestion des Smart Pointers, piège

- Principe des « Smart Pointers ».
- Controverse.
- Opérateurs de conversion de types.

#### Atelier

*Implémentation d'un « smart pointer » avec compteur de références.*

---

### Les templates, patron de classe

- Principe et présentation.
- Syntaxe des classes et fonctions templates.
- Vérification de types.
- Implémenter une classe template : « TStack ».
- Présentation d'une hiérarchie à base de templates (STL, ATL ...).

#### Atelier

*Réalisation d'une classe template « Tableau ».*

---

### Debug en C++

- Notions système pour déboguer en C++.
- Utilisation avancée du débogueur MSDEV (Visual C++). Assembleur, pile. Point d'arrêt conditionnel et points d'arrêt spéciaux. Fonctions étendues.
- Détection de fuites mémoire.
- Détection des violations d'accès à la mémoire.
- Problèmes liées aux méthodes virtuelles.
- Effets de bord.

#### Atelier

*Etudes de cas de différents problèmes.*

---

### Les règles C++

- Pourquoi des règles de programmation.
- Liste des règles.

#### QCM de fin de stage

*Validation des connaissances acquises.*

