

## Formation C++ : Fondamentaux

<b>Durée :</b>	5 jours
<b>Public :</b>	Tous
<b>Pré-requis :</b>	Notions de C ou d'un langage proche
<b>Objectifs :</b>	Apprendre la programmation orientée Objet avec C++ - Etre capable de développer en autonomie avec C++
<b>Sanction :</b>	Attestation de fin de stage mentionnant le résultat des acquis
<b>Taux de retour à l'emploi:</b>	Aucune donnée disponible
<b>Référence:</b>	PRO71-F
<b>Note de satisfaction des participants:</b>	4,50 / 5

### Introduction

Historique, versions et standardisation  
C par rapport au C++  
C++ et les autres langages  
Environnement, outils

### Généralités et rappels

La compilation  
Organisation d'une application en C++  
Grammaire locale (variables, structures, boucles...)  
Fonction main()  
Fonctions standards d'E/S

### Spécificités du C++

Définition de variables, utilisation  
Types principaux et environnement d'exécution  
Formats d'affichage  
Transtypage  
Arithmétique et opérations diverses ; les conditions  
Constantes typées  
Structures conditionnelles  
Boucles et traitement de données multiples  
Tableaux - création, manipulation  
Pointeurs, l'accès à une mémoire quelconque  
Références  
Enumérations

## **Les fonctions**

- Prototypage et définition
- Arguments simples et utilisation
- Passage de paramètres (par valeur, par adresse, par référence)
- Références et valeurs de retour
- Surcharge de fonctions
- Fonctions inline
- Récurtivité
- Organisation des fichiers sources

## **Les structures - introduction à la Programmation Orientée Objet**

- Définition, création, utilisation
- Définition de méthodes
- Déclaration d'une variable de structure
- Structure constante, le mot clé "mutable"
- Structures comme paramètres de fonctions

## **Programmation Orientée Objet**

- Apports de l'objet sur le procédural
- Définition de classes
- Utilisation : instanciation statique ou dynamique
- Attributs et méthodes d'instances
- Constructeurs, destructeurs, surcharge
- Champs statiques et méthodes de classe
- Encapsulation des données (accesseurs et modifieurs)
- L'agrégation, relations entre les données - problèmes et solutions
- Fonctions et classes amies d'une classe (friend)
- Héritage, la réutilisabilité
- Polymorphisme et abstraction des classes
- Héritage multiple : possibilité, problèmes, conseils

## **La surcharge des opérateurs**

- Les besoins, les possibilités, alternatives
- Le mécanisme général
- Recherche d'une solution idéale

## **Programmation générique (Templates)**

- Intérêt - quand utiliser, quand éviter ?
- Templates de fonction et exemples fondamentaux
- Templates de classes
- Instanciation de templates

## **Les espaces de nommage en C++**

- Définition, les namespaces nommés
- Alias d'espace de nommage
- Déclaration Using / Directive Using namespace

## **Les exceptions**

Introduction

Principe et mécanismes de gestion des exceptions

Libération des ressources dynamiques lors d'une exception

## **Bibliothèques standards du C++**

Les conventions du C++, la documentation

Accès aux fichiers

Les chaînes de caractères - aspects complexes des conversions

La STL - conteneur, itérateurs, ...

Présentation des bibliothèques de présentation (SDL, API Win32, MFC, .NET)