

# Mise en oeuvre du langage c ++

## INFORMATIONS GÉNÉRALES

**Type de formation :** Formation continue

**Éligible au CPF :** Non

**Domaine :** Développement

**Action collective :** Non

**Filière :** Action collective ATLAS "Java, .Net, C++"

**Rubrique :** C++

**Code de formation :** F28029

## € Tarifs

**Prix public :** 2275 €

### Tarif & financement :

Nous vous accompagnons pour trouver la meilleure solution de financement parmi les suivantes :

- **Le plan de développement des compétences** de votre entreprise : rapprochez-vous de votre service RH.
- **Le dispositif FNE-Formation.**
- **L'OPCO** (opérateurs de compétences) de votre entreprise.
- **Pôle Emploi** sous réserve de l'acceptation de votre dossier par votre conseiller Pôle Emploi.
- **CPF -MonCompteFormation**

Contactez nous pour plus d'information

## PRÉSENTATION

### Objectifs & compétences

Connaître les concepts de la programmation orientée objet  
Maîtriser la syntaxe du langage C++  
Savoir développer une application C++ d'accès aux fichiers

### Public visé

Développeur d'applications Chefs de projets

### Pré-requis

La connaissance du langage C est fortement conseillée

## 📍 Lieux & Horaires

**Campus :** Ensemble des sites

**Durée :** 35 heures

### Délai d'accès :

Jusqu'à 8 jours avant le début de la formation

**Distanciel possible :** Oui

## PROGRAMME

### Chapitre 1 : Approche objet

Inconvénients de l'approche procédurale  
Origines de l'approche objet  
Encapsulation  
Abstraction  
Héritage  
Polymorphisme  
Panorama des principaux langages objet

### Chapitre 2 : Le langage C++

Caractéristiques et historique C++ vis à vis de C  
Déclaration en C++ Aperçu des E/S standard  
Déclarations de const  
Pointeurs et constantes  
Accès aux variables globales Structures, énumérations, unions et classes  
Type de données void La notion de référence  
Exercices : mise en oeuvre de variables, de branchements conditionnels, de boucles, de pointeurs, de références

### Chapitre 3 : Structure d'un programme C++

Les espaces de noms  
Les fonctions  
Les fonctions in line  
Surcharge de fonctions  
Paramètres en nombre variable  
Exercices : mise en oeuvre de fonctions surchargées

### Chapitre 4 : Le pré-compilateur et compilation séparée

Principe de la compilation séparée  
Rôles de fichiers d'entête L'inclusion de fichiers  
Principe de la compilation conditionnelle  
Les constantes  
Les macro-instructions  
Exercices : mise en oeuvre de la compilation conditionnelle dans les fichiers d'entête et les

## 📅 Prochaines sessions

Cliquez sur la date choisie pour vous inscrire :

■ 07 / 10 / 2024

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

■ 09 / 12 / 2024

📍 : Ensemble des sites

✓ : Distanciel possible

🕒 : 35 heures

📅 : 5 jours

fichiers sources

### **Chapitre 5 : Les classes et les objets**

Généralités  
Déclaration de classe  
Les membres de la classe  
Contrôle d'accès aux membres  
Constructeur et destructeur  
Instanciation  
Accès aux attributs et méthodes  
Membres spéciaux : this  
Les membres amis  
Les membres static  
Les méthodes const  
Classes composées d'objets  
Destruction des instances  
Exercices : création de classes simples, instanciation, ajout de constructeurs et du destructeur, de membres static

### **Chapitre 6 : Concepts avancés**

Rôle du constructeur de copie  
Surcharge de l'opérateur d'affectation  
Exercices : ajout d'un constructeur de copie et surcharge de l'opérateur d'affectation dans la classe précédemment développée

### **Chapitre 7 : La surcharge des opérateurs**

Sommaire des opérateurs  
Conversion de type implicite  
Fonction amies pour la surcharge d'opérateurs  
Opérateur de conversion de type  
Exercices : ajout d'opérateurs surchargés comme méthodes et comme fonctions amies

### **Chapitre 8 : L'héritage et le polymorphisme**

Types d'héritage  
Mécanisme d'héritage  
Classes dérivées  
Hiérarchie de classes  
Polymorphisme et méthodes virtuelles  
Héritage multiple  
Classes de base virtuelles  
Exercice : mise en oeuvre d'un héritage public simple entre trois classes  
Exercice : mise en oeuvre du polymorphisme  
Exercice : mise en oeuvre d'un héritage multiple répété

### **Chapitre 9 : Gestion des exceptions**

Gestion sur place avec try, catch, finally Propagation  
Lever une exception avec throw  
Exercice : mise en place d'une gestion d'exceptions pour traiter des anomalies lors du passage de paramètres  
Complément d'E-Learning avec Lynda sur la gestion des exceptions

### **Chapitre 10 : Introduction aux pointeurs intelligents**

Inconvénients des pointeurs nus  
Principe des pointeurs smart pointers  
Mise en oeuvre de unique\_ptr et shared\_ptr  
Exercice : mise en oeuvre de pointeurs intelligents pour remplacer des pointeurs nus  
Complément d'E-Learning avec Lynda sur les pointeurs intelligents

### **Chapitre 11 : Introduction aux templates Principe des templates**

Instanciation des templates template de fonctions template de classes  
Exercice : création et utilisation d'un template de fonction  
Exercice : création et utilisation d'un template de classe simple  
Complément d'E-Learning avec Lynda sur les templates

### **Chapitre 12 : La bibliothèque STL**

Présentation de la Standard Template Library STL  
Présentation de quelques conteneurs  
Présentation des templates d'entrées/sorties  
Exercice : mise en oeuvre du conteneur std::vector  
Exercice : mise en oeuvre des templates d'entrées-sorties sur fichiers

## MODALITÉS

### Modalités

**Modalités** : en présentiel, distanciel ou mixte – Horaires de 9H à 12H30 et de 14H à 17H30 soit 7H – Intra et Inter entreprise

**Pédagogie** : essentiellement participative et ludique, centrée sur l'expérience, l'immersion et la mise en pratique. Alternance d'apports théoriques et d'outils pratiques.

**Ressources techniques et pédagogiques** : Support de formation au format PDF ou PPT Ordinateur, vidéoprojecteur, Tableau blanc, Visioconférence : Cisco Webex / Teams / Zoom

**Pendant la formation** : mises en situation, autodiagnosics, travail individuel ou en sous-groupe sur des cas réels

### Méthode

**Fin de formation** : entretien individuel

**Satisfaction des participants** : questionnaire de satisfaction réalisé en fin de formation

**Assiduité** : certificat de réalisation (validation des acquis)