

# Programmation Objet en C++

PROGOBC++

Présentiel - Synchrone



## Public Visé

Développeurs, ingénieurs et chefs de projets proches du développement



## Pré Requis

Bonnes connaissances d'un langage de programmation de type C, Java, C#, VB.NET ou PHP

## Objectifs pédagogiques et d'évaluation

À l'issue de la formation l'apprenant sera capable de :  
Maîtriser la syntaxe du langage C++  
Mettre en œuvre les concepts de la Conception Orientée Objet  
Utiliser les outils de développement associés au langage C++  
Maîtriser les ajouts majeurs de la norme C++ 11

## Méthodes pédagogiques

Formation axée sur la mise en pratique  
Méthode participative  
Alternance de cours et d'exercices dirigés ou en autonomie  
Tous les exercices comportent une phase d'analyse/conception suivie d'une phase de programmation  
Evaluation croisée et partage de bonnes pratiques  
Support de cours

## Parcours pédagogique

### La syntaxe du C++ (différences entre C et C++)

Données : définition, initialisation, types de données  
Expressions : notion de référence, mécanismes de cast  
Opérateurs (: :, new, delete)  
Fonctions (passage de paramètres et valeur de retour par référence, valeurs par défaut, inlining, surcharge)  
Utilisation du code C dans un programme C++  
Les références (arguments et valeurs de retour)  
Les types constants  
Les espaces de nommage  
Le typage "automatique" avec le mot-clé auto (C++ 11)  
*Mise en pratique : Prise en main de l'environnement de développement et programmation d'un programme simple.*

### Approche Orientée Objet

Les principes généraux des techniques Objet  
C++ et la programmation Objet  
Une introduction aux méthodologies orientées Objet  
Une introduction aux modèles et à la notation UML (modèle statique, dynamique, modèle de coopération, scénario)  
*Mise en pratique : Application des concepts à une étude de cas.*

### Les classes et les objets C++

Les aspects syntaxiques : les champs, les méthodes, les constructeurs  
Le contrôle d'accès  
L'autoréférence  
Les champs et méthodes statiques  
Les fonctions  
Les méthodes et les classes Friend  
La création dynamique des tableaux d'objets  
Les aspects méthodologiques : la conception des classes  
Les constructeurs de copie et de déplacement (C++11)  
La délégation de constructeurs (C++ 11)  
Introduction aux problématiques de gestion mémoire (pile, tas, ramasse-miettes...)  
*Mise en pratique : La programmation de l'étude de cas. La conception et la construction d'une hiérarchie de classes et d'interfaces.*

### Dérivation et héritage

Principe de la dérivation  
Les aspects syntaxiques : la définition des classes dérivées, les constructeurs  
Le contrôle d'accès  
La mise en œuvre du polymorphisme : les fonctions virtuelles

La réutilisation de code : les classes abstraites  
Les interfaces  
La dérivation multiple  
Les aspects sémantiques et méthodologiques : la factorisation du code  
*Mise en pratique : La mise en place du polymorphisme.*

#### Les exceptions

Les aspects syntaxiques : les blocs de try, la génération des exceptions  
Les aspects méthodologiques : la construction d'une hiérarchie d'exception, l'utilisation des exceptions  
*Mise en pratique : L'introduction des exceptions.*

#### La surcharge des opérateurs

Principe de la surcharge  
Surcharge des opérateurs binaires  
Surcharge particulière : l'opérateur indice, fonction, conversion  
Surcharge des opérateurs de gestion mémoire  
Surcharge des opérateurs '<<' et '>>'  
**Mise en pratique : La surcharge de quelques opérateurs simples.**

#### Les modèles

Modèle de classe  
Principes et mécanismes généraux  
Surcharge des modèles et redéfinition de méthodes  
Modèle de fonction  
Principes et mécanismes généraux  
Surcharge des modèles  
Modèles et surcharge des opérateurs  
Les modèles et les mécanismes de dérivation  
Les améliorations proposées par C++ 11  
*Mise en pratique : Exercices sur les modèles.*

#### Les I/O et aperçu sur la STL

Les I/O  
Le principe des streams et la hiérarchie des classes d'entrée/sortie  
Description de quelques classes d'entrées/sorties  
Aperçu sur la STL  
Objectifs et principes  
Descriptions de quelques modèles et classes  
Les conteneurs, les itérateurs, la boucle basée sur un intervalle (C++ 11)

#### Synthèse

Cycle de vie du logiciel : test, intégration, méthode de mise en production  
Interaction avec les autres environnements  
Analyse critique du C++  
Evolution du C++

*Dernière mise à jour : 03/2022*

## Qualification Intervenant·e·s

Formateur expert en développement et programmation



### Méthodes et modalités d'évaluation

Questionnaire de positionnement Evaluation en cours de formation :  
Cas pratiques Evaluation de fin de formation : Qcm Bilan individuel  
des compétences acquises Questionnaire de satisfaction à chaud  
Attestation de fin de formation

## Modalités d'accessibilité handicap

Nous consulter



### Durée

35.00 Heures

5

Jours

### Effectif

De 3 à 8 Personnes