

## C++ - Programmation Objet

Mettre en oeuvre les principes fondamentaux de la conception orientée objet et de concevoir des applications en C++

★★★★★ 4,8/5 (10 avis)

 Présentiel ou en classe à distance



5 jours (35 h)

Prix inter : 3.090,00 € HT  
Forfait intra : 8.990,00 € HT

Réf.: LA200

**C++ est un langage incontournable pour réaliser des application** évolutives, robustes et efficaces. Cependant, sa puissance nécessite une maîtrise approfondie du développeur, non seulement de la syntaxe, mais aussi des concepts fondamentaux tels que les **modèles de gestion mémoire**, la **programmation orientée objet**, et la **programmation générique**.

Depuis 2011, avec les **versions C++11, C++14, C++17, et C++20**, le langage a subi d'importantes améliorations en termes de productivité et d'efficacité. Cette formation permettra aux participants d'**appréhender les principes fondamentaux** de la conception objet et de les appliquer de façon effective au travers d'une étude de cas incrémentale qui intégrera de nombreux idiomes C++.

Cette formation prépare à la certification CPA - C++ Certified Associate Programmer.

### A qui s'adresse cette formation ?



#### Pour qui

- Développeurs
- Ingénieurs
- Chefs de projets proches du développement



#### Prérequis

- Connaître les principes de la programmation orientée objet
- Disposer d'une expérience d'un langage de programmation
- **Disposez-vous des connaissances nécessaires pour suivre cette formation ? Testez-vous !**

### Programme

#### 1 - Introduction à la conception orientée objet

- Les paradigmes de développement (procédural, objet, fonctionnel, déclaratif)
- Principes de conception objet : Abstraction, Encapsulation, Héritage et Polymorphisme
- Généricité

#### 2 - Syntaxe et modèle de C++

- Point d'entrée d'une application
- Utilisation de la console, objets d'entrée/sortie (streams)
- Typologie des données et initialisation uniforme (C++11)
- Inférence automatique des types avec le spécificateur auto (C++11)
- Les structures de contrôle de flux (if, switch, for, while) et leurs améliorations (C++17/20)
- Les énumérations et leurs nouveautés (C++11/20)
- Fonctions, paramètres et spécificateurs
- Les fonctions lambda (C++11/14/17)
- Fonctions inline et fonctions constexpr (C++11 à C++20)

- Les tableaux et la classe std::array (C++11)
- Introduction à la bibliothèque standard
- Les modèles mémoire (Data segment, Heap, Stack)
- Constantes, pointeurs et références.
- La classe std::string
- Allocation et désallocation dynamiques
- Allocation et désallocation dynamiques Les espaces de noms (namespaces)
- Organisation du code au sein d'un projet (fichiers d'entête et fichiers d'implémentation)

### 3 - Les concepts objet appliqués au C++

- Classes, instances, cycle de vie et cinématique d'un objet
- Éléments de notation UML (modélisation statique et dynamique)
- Encapsulation et visibilité (public-private-protected)
- Méthodes, constructeurs, destructeurs
- La zone d'initialisation des membres (ZIM)
- Fonctions et classes amies (friendship)
- Surcharge (overloading) des méthodes et opérateurs
- Membres de classe (ou statiques)
- Mise en oeuvre des relations (agrégation / composition)
- Robustesse via les spécificateurs de déclaration (default, delete, override, final) de C++11
- Héritage, classes abstraites et concrètes
- Polymorphisme et interfaces
- Destructeur virtuel
- Héritage privé et protégé
- Clonage d'objets (construction par recopie / opérateur d'affectation)

### 4 - Robustesse dynamique et traitement des exceptions

- Prise en compte des erreurs avec les assertions et les exceptions
- Lancement (throw), propagation et interception d'une exception via un gestionnaire (bloc try/catch)
- Conception d'une classe d'exception personnalisée
- Contrôle dynamique d'une exception avec la clause noexcept de C++11

### 5 - Points clés de la Standard Template Library (STL)

- Présentation des principaux conteneurs (vector, list, set, map, deque) et des critères de choix
- Insertion efficace via l'emplacement (C++11)
- Les itérateurs
- Les algorithmes génériques et leurs nouveautés (C++17/20)

### 6 - Généricité - Templates

- Définitions de patrons de fonctions et de classes génériques, syntaxe et instanciation
- Spécialisation générique
- Polymorphisme statique versus polymorphisme dynamique
- Introduction à la métaprogrammation

### 7 - Qualité logicielle et tests d'applications

- Les best practices en C++
- Idiomes et règles de conception
- Introduction aux design patterns
- Le développement guidé par les tests (TDD – Test Driven Development)
- Présentation succincte de GoogleTest



### Les objectifs de la formation

- Appliquer les principes de la Conception Orientée Objet
- Maîtriser la syntaxe du langage C++
- Concevoir des applications C++ utilisant des classes
- Utiliser les outils de développement associés au langage C++
- Maîtriser les ajouts majeurs de la norme C++



## Evaluation

- Pendant la formation, le formateur évalue la progression pédagogique des participants via des QCM, des mises en situation et des travaux pratiques. Les participants passent un test de positionnement avant et après la formation pour valider leurs compétences acquises.



## Les points forts de la formation

- Une pédagogie efficace : l'alternance d'exposés théoriques et de mises en application immédiates à travers de nombreux travaux pratiques.
- Les méthodes, bonnes pratiques et astuces pour tirer le meilleur parti de la programmation en C++ et développer ainsi des applications stables et performantes.
- 93% des participants à cette formation se sont déclarés satisfaits ou très satisfaits au cours des 12 derniers mois.



## Dates et villes 2026 - Référence LA200



Dernières places disponibles



Session garantie

### A distance

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

du 30 mars au 3 avr.

du 20 juil. au 24 juil.

du 7 déc. au 11 déc.

### Toulouse

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

### Aix-en-Provence

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

### Strasbourg

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

### Sophia Antipolis

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

### Lille

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

## **Marseille**

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

## **Rouen**

du 19 janv. au 23 janv.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

## **Paris**

du 30 mars au 3 avr.

du 20 juil. au 24 juil.

du 7 déc. au 11 déc.

du 1 juin au 5 juin

du 28 sept. au 2 oct.

## **Nantes**

du 30 mars au 3 avr.

du 20 juil. au 24 juil.

du 28 sept. au 2 oct.

## **Rennes**

du 30 mars au 3 avr.

du 20 juil. au 24 juil.

du 28 sept. au 2 oct.

## **Bordeaux**

du 30 mars au 3 avr.

du 20 juil. au 24 juil.

du 7 déc. au 11 déc.

## **Lyon**

du 30 mars au 3 avr.

du 20 juil. au 24 juil.

du 7 déc. au 11 déc.