

Mis à jour le 06/06/2025

S'inscrire

Formation C++ Programmation Objet

5 jours (35 heures)

Présentation

Notre formation en C++ programmation Objet vous permettra de maîtriser les principes fondamentaux du développement objet tout en découvrant les spécificités avancées du langage C++. Chaque jour, vous apprendrez à structurer votre code de manière claire, performante et évolutive.

Découvrez notre formation en Programmation Orientée Objet en C++, pensée pour maîtriser progressivement les fondamentaux essentiels du langage. Vous évoluerez étape par étape, depuis les bases jusqu'aux concepts avancés, tout en renforçant constamment vos compétences.

Commencez par une immersion claire dans la syntaxe fondamentale du C++ et les normes récentes, en apprenant toutes les subtilités de la programmation orientée objet ainsi que les bonnes pratiques de factorisation des principes SOLID et de Clean Code.

Au fil de la formation, vous maîtriserez les structures de contrôle, les types de données et la gestion mémoire basique à avancée, essentielles pour écrire des programmes robustes et performants.

Appliquez concrètement les principes de la Conception Orientée Objet : classes, encapsulation, héritage et polymorphisme. Chaque atelier pratique vous permettra de développer progressivement vos compétences techniques sur ce langage moderne. Vous découvrirez également son riche écosystème au travers des bibliothèques populaires, telles que Boost et Poco.

Enfin, en intégrant progressivement les innovations modernes du C++ (normes C++17 à C++23), vous acquerez des compétences immédiatement applicables en milieu professionnel, assurant votre capacité à concevoir des applications durables dans le temps.

Notre programme couvre l'ensemble des notions essentielles du C++, depuis les bases conceptuelles et la syntaxe fondamentale jusqu'à l'ouverture vers des pratiques avancées issues des normes modernes du langage. Vous serez capable de concevoir des applications

robustes, gérer efficacement la mémoire et appliquer des design patterns.

Comme toutes nos formations, celle-ci inclut de nombreux exercices pratiques, celle-ci vous présentera la toute dernière version de C++, à savoir [C++ 23](#).

Objectifs

- Comprendre la syntaxe et les concepts fondamentaux du C++
- Maîtriser les ajouts majeurs des normes C++
- Appliquer les principes de la Conception Orientée Objet
- Écrire des programmes simples en appliquant les bonnes pratiques de développement
- Utiliser les structures de contrôle et les types de données en C++
- Manipuler les fichiers et la mémoire de manière basique

Public visé

- Développeurs,
- Ingénieurs,
- Chefs de projets développement

Pré-requis

- Connaître les principes de la programmation orientée objet et disposer d'une expérience C++

Programme de la formation C++ Objet

Jour 1 : Introduction approfondie et syntaxe essentielle

Découvrir le C++ moderne

- Historique et évolutions récentes
- Présentation générale du langage C++
- Installation et configuration de l'environnement
- Écosystème (IDE, gestionnaires de paquets)
- Rappels sur les bonnes pratiques de développement (Factorisation, Clean Code, SOLID, KISS, etc.)
- Normes ISO récentes
- Atelier pratique : Initialisation du projet ProBank avec compilation basique.

Normes majeures

- C++98/03 : Première standardisation officielle. STL initiale.
- C++11 : Smart pointers, lambdas, auto, constexpr, multithreading.
- C++14 : Améliorations constexpr, auto amélioré, fonctions génériques lambda.

- C++17 : optional, variant, any, Structured bindings, filesystem standard, algorithmes parallèles.
- C++20 : Modules, coroutines, concepts. Ranges et vues avancées.
- C++23 : std::print, std::expected, évolutions sur les modules, nouvelles vues (views::zip, etc.).
- C++26 : Prochaines normes attendues (Réflexion, Contrats, Pack indexing, constexpr avancé, internationalisation améliorée)

Types fondamentaux et avancés

- Types primitifs et dérivés
- Portée et durée de vie
- Const et constexpr
- Auto et déduction de types
- Enum class et types forts
- Opérateurs courants
- Atelier pratique : Définition et gestion des premières données clients.

Maîtriser les structures de contrôle

- Conditions avancées (if constexpr, switch moderne)
- Boucles classiques et modernes
- Boucles range-based (C++11+)
- Gestion de flux (break, continue)
- Pattern matching basique
- Atelier pratique : Implémentation validation robuste des saisies.

Les basiques de la POO en C++ : Classes et Objets

- Création et gestion des classes et des objets
- Encapsulation
- Constructeurs, destructeurs et surcharge des constructeurs
- Notions d'héritage simple et multiple
- Polymorphisme fonctions virtuelles
- Méthodes statiques
- Membres statiques
- Cas pratique : Premiers pas pour structurer les responsabilités et données d'un CompteBancaire

Jour 2 : Modularité et gestion mémoire experte

Fonctionnalités avancées des fonctions

- Namespaces
- Fonctions classiques et inline
- Passage par référence vs valeur
- Valeurs par défaut
- Surcharge de fonctions
- Fonctions récursives
- Lambda expressions et capture
- Atelier pratique : Modularisation avancée des fonctionnalités bancaires.

Gestion avancée de la mémoire

- Allocation dynamique
- Pointeurs et pointeurs intelligents
- Stack vs Heap
- Références et const correctness
- Resource Acquisition Is Initialization
- RAII et gestion automatique
- Prévention et détection des fuites mémoire
- Atelier pratique : « Chasseur de fuites mémoire » corriger un programme complexe contenant volontairement des erreurs mémoire cachées

Entrées/Sorties et gestion avancée des fichiers

- Lecture et écriture des fichiers texte
- Flux de fichiers et buffers
- Gestion d'erreurs d'IO
- Sérialisation basique d'objets dans des fichiers
- Traitement des fichiers CSV/XML basique
- Optimisation des accès disque et bonnes pratiques sur les performances
- Atelier pratique : Sauvegarde et restauration optimisée des données clients.

Jour 3 : Approfondissement de la Programmation Orientée Objet

Classes et encapsulation avancée

- Déclaration et encapsulation stricte
- Constructeurs et destructeurs
- Membres statiques et méthodes const
- Mutateurs et accesseurs
- Pattern constructeur
- Atelier pratique : Structuration avancée classe CompteBancaire.

Héritage avancé et polymorphisme dynamique

- Héritage multiple simple
- Polymorphisme et classes abstraites
- Méthodes virtuelles et override
- Gestion d'interfaces (classe abstraite)
- Dynamic_cast et gestion polymorphique
- Atelier pratique : Création hiérarchisée des comptes courants et épargnes.

Patron de conception en C++ (Design Pattern)

- Avantages en maintenance et évolutivité
- Les 3 grandes familles de Design Patterns
 - Créatrice
 - Structurale
 - Comportementale
- Les principaux Patterns utiles en développement logiciel

- Patterns de création Factory, Singleton
- Patterns de structuration Adapter, Decorator
- Patterns de comportement Observer, Strategy
- Bonnes pratiques de conception
- Atelier pratique : Utiliser les patrons Singleton et Factory pour gérer les comptes bancaires.

Jour 4 : Techniques avancées, STL et bibliothèques populaires

Gestion robuste des exceptions

- Exception standard
- Personnalisation des exceptions
- Bonnes pratiques de capture
- Clean-up automatisé
- Erreurs métiers personnalisées
- Atelier pratique : Gestion avancée exceptions sur transactions.

Programmation générique approfondie

- Templates fonctions/classes
- Spécialisations et optimisations
- Concepts (C++20)
- Contraintes et performances
- Avantages en développement logiciel
- Atelier pratique : Template générique pour gestion des transactions multiples.

STL : Maîtrise des conteneurs et algorithmes avancés

- Vectors, Lists, Maps, Sets avancés
- Algorithmes complexes (sort, transform)
- Itérateurs et ranges (C++20)
- Performances et choix des conteneurs
- Lambdas et STL
- Atelier pratique : Implémentation performante gestion historique transactions.

STL : Maîtrise des conteneurs et algorithmes avancés

- Boost (filesystem, algorithm)
- Poco (réseau, sécurité)
- Avantages pratiques
- Intégration rapide
- Cas d'usage typiques
- Atelier pratique : Utilisation de Boost.Filesystem pour gérer les fichiers bancaires

. Jour 5 : Modernité en C++ (C++17 à C++23), Multiplateforme et Internationalisation

Techniques modernes C++17

- Structured Bindings
- Optional, variant, any
- Filesystem standard
- Parallel algorithms
- Fonctions constexpr
- Atelier pratique : Modernisation avancée du code existant de ProBank.

Innovations C++20 et C++23

- Modules et partitionnement
- Coroutines asynchrones
- Ranges et vues avancées
- Nouveautés C++23 (std::print, expected)
- Atelier pratique : Intégration expérimentale des modules et coroutines dans ProBank.

Gestion multiplateforme et internationalisation

- Compilation multiplateforme (Windows, Linux, macOS)
- Gestion des dépendances externes
- Internationalisation (gettext, ICU)
- Encodages et UTF-8
- Gestion des différences systèmes
- Atelier pratique : Adapter ProBank pour une utilisation internationale et multiplateforme.

Pour aller plus loin

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.