

LANGUAGE C++

Le langage de programmation C++ reste au cœur des technologies modernes, offrant des solutions robustes pour des projets variés allant des systèmes embarqués aux applications complexes. Cette formation, basée sur la dernière version de C++, est conçue pour doter le participant de compétences approfondies, tout en lui permettant d'exploiter les dernières fonctionnalités et bonnes pratiques du langage.



Elle s'adresse aux développeurs souhaitant approfondir leurs connaissances en C++ et s'aligner sur les standards actuels du marché. Elle constitue un véritable atout pour évoluer dans des secteurs techniques et exigeants.

OBJECTIFS

- Maîtriser les bases de la programmation en C++ : Types de données, structures conditionnelles, boucles, fonctions.
- Comprendre les paradigmes avancés : Programmation orientée objet, générique, et fonctionnelle.
- Utiliser efficacement les nouveautés de la dernière version de C++ : Concepts, ranges, modules, et coroutines.
- Optimiser les performances : Gestion fine de la mémoire, compréhension des pointeurs et des références.
- Créer des applications professionnelles : Développement, débogage, et maintenance avec les outils modernes.
- Renforcer la sécurité : Utilisation des meilleures pratiques pour réduire les vulnérabilités.

CONTENU

1. Introduction au C++

- Historique et évolution du langage
- Présentation des outils et IDE (Visual Studio, CLion, etc.)
- Compilation, liens, et exécution

2. Bases du langage

- Syntaxe et structure
- Types de données et opérateurs
- Entrées et sorties standard

3. Programmation orientée objet

- Classes, objets et encapsulation
- Héritage et polymorphisme
- Constructeurs, destructeurs et gestion dynamique

4. Fonctionnalités avancées

- *Templates* et programmation générique
- Exceptions et gestion des erreurs
- Bibliothèque standard (STL)

5. Nouveautés de la dernière version de C++

- Concepts et vérifications statiques
- Modules pour une meilleure modularité
- Ranges et manipulation des collections
- Coroutines pour une programmation asynchrone

6. Optimisation et gestion de mémoire

- Allocation dynamique : *new* et *delete*
- Pointeurs intelligents (*unique_ptr*, *shared_ptr*)
- Techniques de profilage et optimisation

7. Sécurité et meilleures pratiques

- Éviter les erreurs courantes (*buffer overflow*, *memory leaks*)
- Usage sécurisé de la bibliothèque standard
- Méthodologies pour un code maintenable et évolutif

8. Applications pratiques

- Développement d'un projet en équipe
- Intégration des outils de versionnement (Git)
- Mise en œuvre des tests unitaires et fonctionnels

9. Conclusion et certification

- Révision des acquis
- Examen final et validation des compétences
- Certification de participation