Durée :	5 jours soit 35 heures	Référence :	IF-C++FO

Public visé :

Cette formation est destinée aux développeurs et aux ingénieurs de développement.

Pré-requis :

Pour suivre cette formation les apprenants doivent :

- avoir pratiquer un langage informatique
- avoir connaissance des concepts objets

Objectifs pédagogiques :

- Ecrire une application en C++ de type console
- Identifier les apports de chaque version C++ 11, 14 et 17
- Organiser les données en mémoire de manière optimale
- Respecter les bonnes pratiques de programmation objet
- Structurer les répertoires et fichiers d'un projet
- Effectuer des opérations d'Entrée/Sortie (E/S)
- Traiter les erreurs et exceptions

Compétences acquises à l'issue de la formation :

- Ecrire une application en C++ de type console
- Identifier les apports de chaque version C++ 11, 14 et 17
- Organiser les données en mémoire de manière optimale
- Respecter les bonnes pratiques de programmation objet
- Structurer les répertoires et fichiers d'un projet
- Effectuer des opérations d'Entrée/Sortie (E/S)
- Traiter les erreurs et exceptions

Modalités pédagogiques :

Session dispensée en présentiel ou téléprésentiel, selon la modalité inter-entreprises ou intra-entreprises sur mesure. La formation est animée par un(e) formateur(trice) durant toute la durée de la session et présentant une suite de modules théoriques clôturés par des ateliers pratiques validant l'acquisition des connaissances. Les ateliers peuvent être accompagnés de Quizz. L'animateur(trice) présente la partie théorique à l'aide de support de présentation, d'animation réalisée sur un environnement de démonstration. En présentiel comme en téléprésentiel, l'animateur(trice) accompagne les participants durant la réalisation des ateliers.

Note relative au programme : Les ateliers, démonstrations, travaux pratiques et scénarios présentés dans ce programme sont fournis à titre d'exemples illustratifs et ne sont pas exhaustifs. Afin de garantir une adéquation permanente avec les évolutions technologiques, le contenu pédagogique ainsi que les environnements de travaux pratiques (TP) sont susceptibles d'être adaptés, ajustés ou mis à jour au cours du temps.

Moyens et supports pédagogiques :

Cadre présentiel

Salles de formation équipées et accessibles aux personnes à mobilité réduite.

- Un poste de travail par participant
- Un support de cours numérique ou papier (au choix)
- Un bloc-notes + stylo
- Vidéoprojection
- Connexion Internet

- Accès extranet pour partage de documents et émargement électronique

Cadre téléprésentiel

Session dispensée via notre solution iClassRoom s'appuyant sur Microsoft Teams.

- Un compte Office 365 par participant
- Un poste virtuel par participant
- Un support numérique (PDF ou Web)
- Accès extranet pour partage de documents et émargement électronique

Modalités d'évaluation et de suivi :

Avant

Afin de valider le choix d'un programme de formation, une évaluation des prérequis est réalisée à l'aide d'un questionnaire en ligne ou lors d'un échange avec le formateur(trice), qui valide la base de connaissances nécessaires.

Les stagiaires disposent également, depuis leur extranet, d'un **outil d'auto-positionnement** leur permettant d'évaluer leur niveau de maîtrise des compétences visées par la formation. Cette auto-évaluation est réalisée avant le démarrage de la formation à l'aide d'un curseur de positionnement allant du niveau le plus faible au niveau le plus élevé.

Pendant

Après chaque module théorique, un ou plusieurs ateliers pratiques permettent de valider progressivement l'acquisition des connaissances. Un quiz peut accompagner les ateliers pratiques afin de mesurer la compréhension des notions abordées.

Après

À l'issue de la formation, les stagiaires réalisent une nouvelle **auto-évaluation** sur les mêmes compétences que celles évaluées avant la formation. Cette comparaison permet de mesurer leur perception de la progression réalisée et de visualiser les compétences développées au cours de la formation.

Lorsque le programme de formation le prévoit, un examen de certification est organisé conformément aux modalités définies par l'éditeur ou le centre de certification (TOSA, Pearson VUE, ENI, PeopleCert…).

Le formateur réalise également une **évaluation individuelle des compétences** de chaque stagiaire sur l'ensemble des compétences visées par la formation. Pour chacune d'elles, il indique si la compétence est :

- Acquise ;
- En cours d'acquisition ;
- Non acquise.

Lorsque la compétence est évaluée comme **en cours d'acquisition** ou **non acquise**, le formateur formule des observations précisant les difficultés rencontrées ainsi que les actions ou recommandations permettant au stagiaire de poursuivre sa montée en compétences.

Enfin

Un questionnaire de satisfaction permet au participant d'évaluer la qualité de la prestation de formation.

Informations sur l'accessibilité :

Nos formations sont, dans la mesure du possible, conçues pour être accessibles à toutes et à tous. Afin de garantir les meilleures conditions d'accueil et d'apprentissage pour les personnes en situation de handicap, nous vous invitons à contacter notre référente handicap certifiée :
Céline SOLATGES – 05 61 34 39 80 – csolatges@iform.fr
Nous vous remercions de bien vouloir nous communiquer toute information utile à ce sujet en amont de la formation, afin de mettre en place les adaptations nécessaires et d'assurer un accompagnement optimal.

Pour en savoir plus sur les dispositifs d'accompagnement existants, vous pouvez consulter les sites suivants :

- AGEFIPH
- FIPHFP
- MDPH de votre département

Description / Contenu

Module 1 : Prise en main

- Processus de compilation

- Atouts et évolutions du langage
- Premier programme
- Affichage de données

Module 2 : Variables et instructions

- Déclarations, typages et constantes
- Opérateurs arithmétiques, booléens et binaires
- Structures conditionnelles : if, switch, ternaire
- Boucles : for, while, do-while, break, continue
- Fonctions et paramètres
- Modules : .h/.cpp

Module 3 : Structures de données

- Enumérations
- Structures et unions
- Tableaux
- Passage de paramètres
- Allocation dynamique : new, delete
- Pointeurs : déclaration, manipulation

Module 4 : Classes et héritage

- Classes et instances
- Membres : public et private
- Constructeur/destructeur
- Surcharge d'opérateurs
- Héritage : syntaxes, protected
- Polymorphisme : virtual

Module 5 : Entrées/Sorties

- E/S standard : cin, cout, cerr
- Formatage
- Surcharge de << et >>
- Fichiers

Module 6 : Gestion d'erreurs

- Limite des codes d'erreur
- Assertions : assert
- Exceptions : throw
- Traitement : try, catch
- Exceptions personnalisées : exception, what()

Module 7 : STL

- Chaînes
- Syntaxe des templates
- Conteneurs : vector, list, deque
- Itérateurs
- Tables de hachage : map, set