CROSSTHINK

Programmation Objet en C++

Ce stage intensif a deux objectifs : initier les participants aux méthodes et réflexes de la programmation par objets ; leur apporter une maîtrise opérationnelle complète du langage C++. Il est le fruit d'une longue expérience en matière de développement en C++ et est construit selon une pédagogie rigoureuse reposant sur des travaux pratiques nombreux et progressifs.

Public et pré-requis

Cette formation ne nécessite pas de prérequis.

Objectifs

•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

Moyens pédagogiques

•Présentation du formateur et du programme.
•Présentation et écoute de chacun de stagiaires.
•Apports didactiques pour apporter des connaissances communes.
•Mises en situation de réflexion sur le thème du stage et des cas concrets.
•Méthodologie d'apprentissage interactive et participative.
•Exercices et études de cas concrets.
•Temps d'échanges.
•Accompagnement pédagogique individualisé.

Formateur

Les formateurs de CROSSTHINK sont des experts de leur domaine, disposant d'une expérience terrain qu'ils enrichissent continuellement. Leurs connaissances techniques et pédagogiques sont rigoureusement validées en amont en interne.

Suivi de l'execution et évaluation des résultats

•Accueil des stagiaires dans une salle dédiée à la formation / A distance
•Feuilles de présence.
•Documents supports de formation projetés.
•Mise à disposition du stagiaire des documents et supports de formation.
•Tout au long et/ou à l'issue de la formation : Evaluation des acquis des stagiaires via des exercices, des QCM, des QUIZZ, des mises en situation et/ou des cas pratiques.
•Enquête de satisfaction.
•Attestation de fin de formation.

Ce stage intensif a deux objectifs : initier les participants aux méthodes et réflexes de la programmation par objets ; leur apporter une maîtrise opérationnelle complète du langage C++. Il est le fruit d'une longue expérience en matière de développement en C++ et est construit selon une pédagogie rigoureuse reposant sur des travaux pratiques nombreux et progressifs.

EN BREF

5 jours

2490€

CONTACT

contact@crossthink.fr

04.83.43.47.48

PROCHAINES DATES

Nous consulter

CROSSTHINK

Programmation Objet en C++

Programme

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).
Travaux pratiques *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).
Travaux pratiques *Application des concepts à une étude de cas qui sera l’un des fils directeurs des exercices suivants.µ*

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…).
Travaux pratiques *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.
Travaux pratiques *La mise en place du polymorphisme dans l’étude de cas.*

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.
Travaux pratiques *L’introduction des exceptions dans l’étude de cas.*

CROSSTHINK

Programmation Objet en C++

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 '>>'.
Travaux pratiques *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.
Travaux pratiques *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).

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

Certification