

Programmation Objet en C++

Cours Pratique de 5 jours - 35h

Réf : C++ - Prix 2024 : 2 970CHF HT

Ce cours 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. Le formateur présentera, durant ces 5 jours, les évolutions des normes de C++98 à C++20.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

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

MÉTHODES PÉDAGOGIQUES

Tous les exercices comportent une phase d'analyse/conception suivie d'une phase de programmation.

TRAVAUX PRATIQUES

Mise à disposition de stations de travail disposant des langages Visual C++ (sous Windows) et gcc (sous Unix). Les TP ont été conçus pour illustrer tous les éléments du langage et pour systématiquement mettre en œuvre les concepts de la conception orientée objet.

LE PROGRAMME

dernière mise à jour : 05/2022

1) 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.

2) 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.

PARTICIPANTS

Développeurs, ingénieurs et chefs de projets proches du développement.

PRÉREQUIS

Bonnes connaissances d'un langage de programmation de type C, Java, Python, C#, VB.NET ou PHP.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques...
Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

- Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
- À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

3) 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.*

4) 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.*

5) 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.*

6) 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.*

7) 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.*

8) 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).

9) 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++.

LES DATES

CLASSE À DISTANCE

2024 : 03 juin, 09 sept., 09 déc.