

C++ avancé: maîtriser les objets (CPPADVFR, 4 jours)

Description

Ce cours certifié SETC offre aux programmeurs d'expérience en C++ les connaissances nécessaires pour créer de meilleures applications. Le cours commence par une brève introduction à la programmation orientée objet, suivi par les notions de fonctions purement virtuelles, de classes abstraites, d'héritage multiple et de polymorphisme. Les concepts de modèles (templates), de modèles de conception (design patterns) et l'utilisation de bons pointeurs sont également couverts dans un esprit pratique menant à leur application. Ce cours présente toutes les notions nécessaires pour que le participant maîtrise l'art et la science de la programmation orientée objet en C++.

Tarifs

- Tarification: \$3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Sujets préparatoires essentiels

Architecture de la mémoire : Le tas et la pile

Variables : Déclaration et instanciation

Variables : l'opérateur d'affectation

Casting et conversion : L'ancien style C par rapport aux opérateurs de casting spécifiques

Variables : Utilisation des objets et des constructeurs

Déclarer les constantes

Utilisation efficace des constantes (et #define)

Le monde des pointeurs et de la gestion de la mémoire

L'architecture des pointeurs

Déclarer des pointeurs sur le tas et sur la pile

Les opérateurs New et Delete

Utiliser Const avec les pointeurs

Utiliser Null et les pointeurs

Opérateurs et pointeurs

Fonctions

Utilisation des fonctions Iostream

La signature d'une fonction

Définition de la portée d'une fonction

Utilisation des fonctions amies

Passage de paramètres par référence ou par pointeur

Passage de paramètres par valeur

Retourner des valeurs à partir d'un appel de fonction

Fonctions en ligne

Concepts et classes orientés objet

Qu'est-ce qu'une programmation orientée objet ?

Définition de la portée d'un membre : Privé, public et protégé

Le cas particulier du constructeur

Définir les constructeurs

Définir les destructeurs

Pointeurs et classes

Utilisation des membres statiques et des fonctions

Comptage des références

Construire une classe simple

Définir des classes efficaces et réutilisables

Conception orientée objet : Une approche générale à l'aide de l'UML

L'objet en C++ : la classe

Définir les variables membres

Encapsulation : La nécessité des variables membres privées

Création de propriétés : Définir les fonctions d'inspecteur

Création d'un objet simple

Création de méthodes

Constructeurs, destructeurs et opérateurs de base

Définition des constructeurs

Définition des opérateurs de conversion

Problèmes de conversion conflictuels

Construire une classe plus compliquée en utilisant le transfert et le comptage de références

Utilisation des listes d'initialisation des membres

Le constructeur par défaut

Le constructeur de copie

La forme canonique

Le destructeur

Surcharge de l'opérateur d'assignation

Surcharge des opérateurs arithmétiques de base (operator+, Operator-, Etc...)

Surcharge des opérateurs postfixe et préfixe

Travailler avec plusieurs objets et plusieurs fichiers

Conception orientée objet

Définition de classes de base abstraites

Héritage et polymorphisme

Définition de la liste d'héritage

Héritage public, privé et protégé

Amitié et héritage

Définition du polymorphisme

Utilisation des fonctions virtuelles

Travailler avec les pointeurs de la classe de base

Construire un exemple de classe multiple

Utilisation des modèles

Qu'est-ce qu'un modèle ?

Problèmes de performance des modèles

Construction d'un modèle de fonction

Structures de données : Tableaux, listes et cartes

Modèles de classe

Construire une liste chaînée à partir de zéro

À propos de la STL et des autres bibliothèques de modèles

Modèles de conception

Aperçu des modèles de conception

À propos des modèles de conception créatifs

À propos des modèles de conception comportementale

À propos des modèles de conception structurels

Sujets avancés

Empêcher la création d'objets sur la pile et le tas

Gestion des exceptions et spécifications des exceptions

Utilisation et création de pointeurs intelligents

Modèles de conception et leur utilisation

Définition d'opérateurs d'allocation de mémoire personnalisés