

Introduction à C++: familiarisation avec les objets (CPPINTFR, 4 jours)

Description

Ce cours certifié SETC offre aux participants tout ce dont ils ont besoin pour commencer la programmation orientée objet utilisant C++. Le cours commence par un survol rapide de la syntaxe de base du langage C++, suivi par les concepts de l'encapsulation et du développement utilisant des classes. Les concepts de données statiques, fonctions virtuelles, héritage et polymorphisme sont également discutés d'un point de vue pratique. À la fin de ce cours, les participants auront maîtrisé les bases de la programmation orientée objet utilisant C++, l'un des langages de programmation les plus flexibles qui existe.

Tarifs

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

Plan de cours

La place du C++ dans la famille étendue des langages de programmation

Code compatible binaire, compatible source et interprété

Visual Basic, C++, Java et langages de scripts

C++ versus C

Points forts et points faibles de C++

Les origines du C++

Le processus de développement du code : La nécessité de planifier

Les meilleures pratiques de l'ingénierie logicielle

Caractéristiques d'une bonne solution logicielle

Comment construire un bon logiciel

Développement itératif

Gestion des exigences

Utilisation d'architectures à base de composants

Vérification continue de la qualité du logiciel

Contrôle des modifications du logiciel

Le processus d'ingénierie logicielle

Le processus unifié Rational

La phase de conception

La phase d'élaboration

La phase de construction

La phase de transition

Le RUP : Structure dynamique

Symptômes et causes profondes des problèmes de développement logiciel

Les cas d'utilisation dans le processus global

Modélisation des processus métier

Les cas d'utilisation dans le processus de développement logiciel

Cas d'utilisation et exigences

Gestion des exigences et des cas d'utilisation

Rédaction des cas d'utilisation

Notation graphique

Formats des cas d'utilisation

Sections des cas d'utilisation

La spécification supplémentaire

Conception avec UML

Exigences fonctionnelles et non fonctionnelles

Passer de l'analyse à la conception

Modélisation de la base de données

Diagrammes d'activité commerciale UML

UML : Diagrammes de classes de conception

UML : Diagrammes de séquence

UML : Diagrammes Statechart

Gestion des grands systèmes

Mise en œuvre d'un processus de gestion des changements

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

Utilisation des constantes avec les pointeurs

Utilisation de null et des pointeurs

Opérateurs et pointeurs

Fonctions

Utilisation des fonctions iostream

La signature d'une fonction

Définir la portée d'une fonction

Utilisation des fonctions amies

Passer des 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 ?

Définir des fonctions efficaces et réutilisables

Conception orientée objet : Une approche générale utilisant UML

L'objet en C++ : la classe

Définir les variables membres

Encapsulation : Le besoin de variables membres privées

Création de propriétés : Définir les fonctions d'inspecteur

Création d'un objet simple

Création de méthodes

Définir la portée d'un membre : private, public et protected

Le cas particulier du constructeur

Définir les constructeurs

Définir les destructeurs

Pointeurs et classes

Utilisation des membres et des fonctions statiques

Comptage des références

Construction d'une classe simple

Constructeurs, destructeurs et opérateurs de base

Définir des constructeurs

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'affectation

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

Surcharge des opérateurs postfixe et préfixe

Définition des opérateurs de conversion

Problèmes de conversion conflictuels

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

Travailler avec plusieurs objets et plusieurs fichiers

Conception orientée objet

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 classe de base

Construction d'un exemple de classes multiples

Définition de classes de base abstraites

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 rien

À propos de STL et des autres bibliothèques de modèles