C++ Certified Senior Programmer – CPS

Objectifs

- Cette **formation C++ Programmation Avancée Expert** vous donne les connaissances et compétences nécessaires pour :
  - Maîtriser les meilleures pratiques concernant l'utilisation des classes
  - Comprendre le besoin et les pièges des conversions (cast)
  - Comprendre l'intérêt du Run-time type information (RTTI)
  - Maîtriser l'utilisation des pointeurs (sur membres, smart pointers, etc.)
  - Programmer efficacement les exceptions
  - Savoir utiliser les templates et les design patterns
  - Gérer un objet qui se comporte comme une fonction (foncteur)
  - Utiliser la puissance de la bibliothèque STL (Standard Template Library)
  - Savoir construire des idiomes
  - Découvrir les possibilités de la bibliothèque Boost
  - Connaître les principales nouveautés du C++11

Prérequis

- Une expérience de développeur en programmation C++ est essentielle pour tirer pleinement profit de cette formation. Avoir suivi la formation C++ Programmation objet en C++ (DPOC) ou posséder un niveau équivalent est un minimum.

Programme

Les classes en C++

Constructeurs et allocation mémoire
Forme canonique d'une classe
Rôle du constructeur de copie
Surcharge de l'opérateur d'affectation
Intérêt d'un destructeur virtuel
Pièges à éviter

Travaux Pratiques :Exécution d'exemples pédagogiques illustrant les concepts présentés

Les conversions en C++

Présentation des conversions, syntaxe
Utilisation du const_cast pour enlever un caractère const
Utilisation du static_cast pour effectuer une conversion standard
Utilisation du reinterpret_cast pour effectuer une conversion forte
Utilisation du dynamic_cast dans une hiérarchie de classesTravaux Pratiques :Choix du type de cast et mise en œuvre dans différentes situations

L’identification de type à l'exécution (RTTI)

Principe et cas d'utilisation
Utilisation du dynamic_cast pour effectuer un downcast dans une hiérarchie de classes
Utilisation de l'opérateur typeid et de la classe type_infoTravaux Pratiques :Mise en œuvre de RTTI pour effectuer un affichage spécifique d'un objet faisant partie d'une hiéarchie de classes

Les pointeurs sur membres de classes

Syntaxe des pointeurs sur membres de classes
Mise en œuvreTravaux Pratiques :Utilisation de pointeurs sur méthodes pour effectuer des calculs mathématiques

Les pointeurs intelligents (smart pointers C++11)

Danger des pointeurs nus
Principe de la gestion de ressources
Intérêt des smart pointers
Raisons de l'obsolescence de auto_ptr
Mise en œuvre de unique_ptr, shared_ptr et weak_ptrTravaux Pratiques :Remplacement, dans une application, de pointeurs nus par des pointeurs intelligents

La gestion des exceptions

Principe des exceptions
Les classes d'exceptions
Comment lever une exception
Gestionnaires d'exceptions
Liste d'exceptions
Hiérarchies d'exceptions
Classes d'exceptions standards
Constructeurs et exceptions
Exceptions et gestion des ressources
Bonnes pratiquesTravaux Pratiques :Mise en place d'une gestion d'exceptions dans une application effectuant des entrées-sorties

Les templates

Présentation
Avantages/inconvénients
Syntaxe des templates de fonctions
Syntaxe des templates de classes
Syntaxe des templates de méthodes
Instanciation des templates de fonctions
Instanciation des templates de classes
Spécialisation partielle ou totale des templatesTravaux Pratiques :Mise en œuvre d'une fonction template
Mise en œuvre d'une classe template
Mise en œuvre d'une classe template template

Introduction aux Design Patterns (avec zoom sur certains)

Présentation des patterns du GoF
Patterns de création
Patterns de structure
Patterns de comportementTravaux Pratiques :Mise en œuvre des patterns Singleton, Factory Method, Abstract Factory
Mise en œuvre des patterns Visitor, Proxy

Les foncteurs

Présentation
Intérêt des foncteurs
Foncteurs prédéfinis dans la bibliothèque standard
Utilisation d'adaptateurs de fonctions unaires et binairesTravaux Pratiques :Mise en œuvre d'un foncteur avec l'algorithme for_each pour afficher le contenu d'un vector

La bibliothèque STL

Présentation de la Standard Template Library
Les conteneurs
Les allocateurs
Les itérateurs
Les algorithmes
Les entrées-sortiesTravaux Pratiques :Mise en œuvre de quelques conteneurs, d'algorithmes et template d'entrées-sorties

Les idiomes

Traits
Policy
SFINAE (Substitution Failure Is Not An Error)
CRTP (Curiously Recurring Template Pattern) : pour le polymorphisme statiqueTravaux Pratiques :Mise en œuvre de chacun des idiomes

La méta-programmation

Comment exécuter à la compilation
Avantages/inconvénients
OptimisationsTravaux Pratiques :Mise en œuvre de la méta-programmation pour effectuer des calculs mathématiques par le compilateur

La bibliothèque Boost

Présentation
static_assert
property_map
smart_ptr
tuple
any

variant
threads
interprocess
mpl (méta programming language)Travaux Pratiques :Mise en œuvre de quelques templates de Boost

Nouveautés essentielles du C++11

Mot-clés auto, decltype et constexpr
Définition des rvalue références
Application des rvalue références : déplacement et transfert parfait
Bonne utilisation de std::move et std::forward
Les fonctions lambda
Les variadic templatesTravaux Pratiques :Mise en œuvre des mot-clés auto, decltype et constexpr
Mise en œuvre des références rvalue pour la création et la copie d'objet par déplacement
Mise en œuvre des expressions lambda en remplacement des foncteurs
Mise en œuvre des variadic templates

Autres nouveautés du C++11

Initialisation des données membres non-statiques
Alias de template
Constructeurs délégués
Déclarations étendues de l'amitié
Surcharge explicite de la virtualité
La constante nullptr
« Range-based » for
Définition des rvalue références
Les opérateurs de conversion explicites
Les types POD (Plain Old Data) revisités
Les types locaux et non nommés comme arguments template
Les énumérations à typage fort
Les fonctions par défaut et supprimées (=default, =delete)
Les espaces de nom inline
La propagation des exceptions (dans le cadre du multithreading)Travaux Pratiques :Mise en œuvre d'une partie de ces nouveautés dans une application existante

Performances

Introduction
Résumé des bonnes pratiques