Design patterns & performance en C++

Référence : CPP-DPP | **Niveau : 2 - Avancé** | **Contact : Virginie Pigeat**

Durée : 3 jours (21h.) | Classe à distance : Possible | 01 69 15 32 32 / 07 87 10 03 92

Tarif : 1 500 € h.t / personne | virginie.pigeat@agenium.com

Objectifs

Cette formation vise à comprendre le cycle de vie des objets en C++.

Connaître les Design Patterns pour écrire un code maintenable et performant. Gagner en performance en écrivant un code parallèle.

Public

Cette formation C++ s'adresse aux développeurs, ayant une bonne connaissance de base du C++.

Pré-requis

Pour suivre cette formation, les participants doivent disposer d'une expérience de programmation en C++.

Travaux pratiques

Il s'agit d'un cours avec une grande partie consacrée aux travaux dirigés (plus de 50%).

Moyens pédagogiques et techniques

Les formations Agenium Campus sont conçues et animés par des experts en activité.

Nos salles sont équipées de vidéoprojecteur et écran/tableau et d'un accès internet. Chaque participant dispose d'un poste de travail et d'un support de cours.

Nos formations peuvent être suivies à distance.

Modalités de suivi et d'évaluation

Les participants signent une feuille de présence par demi-journée. Une attestation de validation des acquis est remise à la fin de la formation.

L'évaluation en cours de formation est réalisée grâce à des exercices ou études de cas (50% du temps minimum pour les cours pratiques) et/ou sous forme de QCM.

L'évaluation en fin de formation Un QCM ou un exercice est donné aux stagiaires après la formation afin de mesurer l'acquisition des connaissances.

Niveau de satisfaction : 5 / 5

Le contenu de nos formations est adaptable selon vos besoins

Enregistrée sous le numéro 117 884 289 78, cet enregistrement ne vaut pas agrément de l'Etat
La certification qualité a été délivrée au titre de la catégorie d'action suivante : ACTIONS DE FORMATION

Qualiopi processus certifié
RÉPUBLIQUE FRANÇAISE

Design patterns & performance en C++

AGENIUM CAMPUS

Programme :

CYCLE DE VIE DES OBJETS

- Valeur, référence et pointeurs
- Transfert
  - Exemple
  - En pratique
  - Exemple de réalisation
  - Contre-exemples
- Optimisation et élision des copies
  - Exemple
  - Conditions
  - Utilité
- Règle du zéro
- Comportements indéfinis
  - Définition
  - Exemples
- Programmation générique

DESIGN PATTERNS

- Présentation
  - Présentation des Design Patterns
  - Critique des Design Patterns
- Notation UML simplifiée
- Méthodologie SOLID
  - Single Responsibility Principe
  - Open/Closed Principe
  - Liskov Substitution Principe
  - Interface Segregation Principe
  - Dependency Inversion Principe
- Design patterns du C++
  - RAII
  - CRTP
- Design patterns généraux
  - Itérateur
  - Observateur
  - Singleton
  - Stratégie
  - Décorateur
  - Visiteur
  - Fabrique
  - Fabrique Abstraite

PERFORMANCE ET PARALLELISME

- Optimisation du compilateur
  - Exemples
  - Coder simple pour aller vite
  - Rappels sur les comportements indéfinis
- Architecture moderne des processeurs
- Array of structure vs. Structure of arrays
- Calculs parallèles
- Design parallèle en C++
  - Accès concurrents
  - Passage de valeurs

Enregistrée sous le numéro 117 884 289 78, cet enregistrement ne vaut pas agrément de l'Etat
La certification qualité a été délivrée au titre de la catégorie d'action suivante : ACTIONS DE FORMATION

Qualiopi processus certifié
RÉPUBLIQUE FRANÇAISE