Programmation C++, perfectionnement

-Réference: **MR-114** -Durée: **4 Jours (28 Heures)**

Les objectifs de la formation

- Découvrir les nouveautés apportées par la version C++11
- Maîtriser la gestion de la mémoire, des pointeurs et des références
- Implémenter la généricité en C++Découvrir la bibliothèque standard STL
- Utiliser la librairie BOOST et C++11

A qui s'adesse cette formation ?

POUR QUI :

- Concepteurs et développeurs d'applications en C++, chefs de projets, architectes logiciels.

Programme

- **Les bibliothèques de fonctions**
  - Bibliothèques standard du langage : ctype.
  - h, math.
  - h, stdlib.
  - h, time.
  - h.
  - et les autres.
  - Bibliothèques mathématiques avancées : Linpack, Lapack.
  - La gestion de l'allocation dynamique : fonctions calloc(), realloc().
  - Fonctions à nombre d'arguments variables existantes et créées par le programmeur.
  - Travaux pratiques Utilisation de plusieurs bibliothèques de fonctions.
- **Rappels**
  - Classes d'allocation mémoire.
  - Construction, initialisation, embarquement d'objets.

Programme

- Les fuites mémoire.
- Constance, le mot-clé mutable, Lazy Computation.
- Amitié (friendship) C++ et contrôle d'accès.
- Destruction virtuelle.
- Stratégie de gestion des exceptions.
- Les espaces de nommage (namespace).

- **Les nouveautés langage de C++11**
  - nullptr et autres littéraux.
  - Les directives =delete, =default.
  - Délégation de constructeurs.
  - Les énumérations «type safe».
  - Le mot-clé auto et boucle sur un intervalle.
  - Référence rvalue et impacte sur la forme normale des classes C++.
  - Les lambda-expressions.
  - Travaux pratiques Réécriture d'un code C++ existant en C++11, comparaison des deux implémentations.

- **Gestion des opérateurs**
  - Opérateurs binaires et unaires.
  - L'opérateur d'indirection, cas d'usage.
  - L'opérateur de référencement.
  - Les opérateurs d'incrémentation/décrémentation préfixés et post-fixés.
  - Les autres opérateurs : comparaison, affectation.
  - La surcharge de l'opérateur [], des opérateurs d'insertion (TR1 --> C++11.
  - Les nouveaux conteneurs : array, forward_list, unordered_set, unordered_map.
  - La classe tuple.
  - Les pointeurs intelligents (smart pointer) : shared_ptr, weak_ptr, unique_ptr.
  - Les nouveaux foncteurs et binders.
  - Introduction à la gestion des threads.
  - Les expressions régulières.
  - Travaux pratiques Mise en oeuvre de la robustesse avec les smart pointers.
  - Utilisation d'expressions régulières.

Programme

- **BOOST**
  - La Pointer Container Library (destruction des données pointées d'un conteneur).
  - Les structures de données boost::any et boost::variant.
  - Programmation événementielle (connexions et signaux).
  - Gestion des processus, mécanismes de communication interprocessus et mémoire partagée.
  - Travaux pratiques Amélioration de l'implémentation de l'étude de cas par l'utilisation la Pointer Container Library.
- **Utilisation avancée de l'héritage**
  - Héritage versus embarquement.
  - Héritage privé.
  - Héritage protégé.
  - Exportation de membres cachés avec la Clause Using.
  - Héritage multiple et gestion des collisions de membres.
  - Héritage en diamant.
  - Héritage virtuel et dynamic_cast.
  - Principes de conception : substitution de Liskov, principe d'ouverture/fermeture, inversion des dépendances.
  - Règles d'implémentation des interfaces en C++.
  - Travaux pratiques Combinaison de l'héritage multiple, privé et de l'exportation pour concevoir des classes robustes et hautement évolutives.


(+212) 5 22 27 99 01


(+212) 6 60 10 42 56


Contact@skills-group.com

Nous sommes à votre disposition :
De Lun - Ven 09h00-18h00 et Sam 09H00 – 13H00

Angle bd Abdelmoumen et rue Soumaya, Résidence Shehrazade 3, 7éme étage N° 30
Casablanca 20340, Maroc