

Programmation objet en C++

-Référence: **SII-298**

-Durée: **5 Jours (35 Heures)**

Les objectifs de la formation

- Maîtrise de la syntaxe en C++
Appliquer les concepts de conception axée sur l'objet
- Utilisation d'outils de développement en lien avec la langue C++
Maîtriser les grandes nouveautés de la norme C++ 11

A qui s'adresse cette formation ?

POUR QUI :

- Développeurs, ingénieurs et chefs de projets proches du développement.

PRÉREQUIS :

- Bonnes connaissances d'un langage de programmation de type C, Java, Python, C#, VB.
- NET ou PHP.
- .

Programme

- **La syntaxe du C++ (différences entre C et C++)**
 - Données : définition, initialisation, types de données.
 - Expressions : notion de référence, mécanismes de cast.
 - Opérateurs (: :, new, delete).
 - Fonctions (passage de paramètres et valeur de retour par référence, valeurs par défaut, inlining, surcharge).
 - Utilisation du code C dans un programme C++.
 - Les références (arguments et valeurs de retour).
 - Les types constants.
 - Les espaces de nommage.
 - Le typage "automatique" avec le mot-clé auto (C++ 11).
 - Travaux pratiques Prise en main de l'environnement de développement et programmation d'un programme simple.

- **Approche orientée objet**

- Les principes généraux des techniques objet.
- C++ et la programmation objet.
- Une introduction aux méthodologies orientées objet.
- Une introduction aux modèles et à la notation UML (modèle statique, dynamique, modèle de coopération, scénario).
- Travaux pratiques Application des concepts à une étude de cas qui sera l'un des fils directeurs des exercices suivants.

- **Les classes et les objets C++**

- Les aspects syntaxiques : les champs, les méthodes, les constructeurs.
- Le contrôle d'accès.
- L'autoréférence.
- Les champs et méthodes statiques.
- Les fonctions.
- Les méthodes et les classes friend.
- La création dynamique des tableaux d'objets.
- Les aspects méthodologiques : la conception des classes.
- Les constructeurs de copie et de déplacement (C++11).
- La délégation de constructeurs (C++ 11).
- Introduction aux problématiques de gestion mémoire (pile, tas, ramasse-miettes).
- Travaux pratiques La programmation de l'étude de cas.
- La conception et la construction d'une hiérarchie de classes et d'interfaces.

- **Dérivation et héritage**

- Principe de la dérivation.
- Les aspects syntaxiques : la définition des classes dérivées, les constructeurs.
- Le contrôle d'accès.
- La mise en oeuvre du polymorphisme : les fonctions virtuelles.
- La réutilisation de code : les classes abstraites.
- Les interfaces.
- La dérivation multiple.
- Les aspects sémantiques et méthodologiques : la factorisation du code.
- Travaux pratiques La mise en place du polymorphisme dans l'étude de cas.

- **Les exceptions**

- Les aspects syntaxiques : les blocs de try, la génération des exceptions.
- Les aspects méthodologiques : la construction d'une hiérarchie d'exception, l'utilisation des exceptions.
- Travaux pratiques L'introduction des exceptions dans l'étude de cas.

- **La surcharge des opérateurs**

- Principe de la surcharge.
- Surcharge des opérateurs binaires.
- Surcharge particulière : l'opérateur indice, fonction, conversion.
- Surcharge des opérateurs de gestion mémoire.
- Surcharge des opérateurs " ".
- Travaux pratiques La surcharge de quelques opérateurs simples.

- **Les modèles**

- Modèle de classe.
- Principes et mécanismes généraux.
- Surcharge des modèles et redéfinition de méthodes.
- Modèle de fonction.
- Principes et mécanismes généraux.
- Surcharge des modèles.
- Modèles et surcharge des opérateurs.
- Les modèles et les mécanismes de dérivation.
- Les améliorations proposées par C++ 11.
- Travaux pratiques Exercices sur les modèles.

- **Les I/O et aperçu sur la STL**

- Les I/O.
- Le principe des streams et la hiérarchie des classes d'entrée/sortie.
- Description de quelques classes d'entrées/sorties.
- Aperçu sur la STL.
- Objectifs et principes.
- Descriptions de quelques modèles et classes.
- Les conteneurs, les itérateurs, la boucle basée sur un intervalle (C++ 11).

Programme

- **Conclusion**

- Cycle de vie du logiciel : test, intégration, méthode de mise en production.
- Interaction avec les autres environnements.
- Analyse critique du C++.
- Evolution du C++.



(+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, 7ème étage N° 30
Casablanca 20340, Maroc