

# Programmation Orientée Objet en C++ – Initiation

## Programme

---

### Historique et introduction à la programmation C++

- Filiation avec le langage C
- Pourquoi C++ ?
- Les différentes versions de la norme C++ ISO et les liens avec Boost
- Le langage C++ ISO 2017
- Présentation de GNU G++ et d'Eclipse/CDT
- Travaux pratiques : mise en oeuvre d'un « Hello World » en C++ avec Eclipse/CDT

### Quelques premières améliorations par rapport à C

- Les extensions de C++ et le standard C99
- Les commentaires en C++
- Une gestion des types plus strictes et le typage automatique
- Le type bool
- Simplification de l'utilisation des types énumérés, des structures et des unions
- Les types énumérés fortement typés
- Le principe de localité
- Les nouveaux opérateurs de gestion dynamique de la mémoire
- Passage de paramètres par références et comparaison avec le mode de passage par pointeurs
- Valeurs de paramètres par défaut
- Surcharge de fonctions et d'opérateurs
- Fonctions inlines
- TP : implémentation progressive d'une librairie mathématiques en utilisant ces différents concepts

### Introduction à la Programmation Orientée Objet et principe d'encapsulation

- Pourquoi utiliser la POO (Programmation Orientée Objet)
- Le vocabulaire propre à la POO
- Comparatif entre structure C et classe C++
- Instanciation et utilisation d'objets
- Notion d'attributs et méthodes
- Protéger des membres de la classe via la notion de visibilité
- Constructeurs et destructeurs d'objet
- Délégation de constructeurs
- Constructeurs par copies
- Le principe d'encapsulation
- Définitions d'opérateurs membres de la classes
- Assistance proposées par Eclipse/CDT
- TP : la librairie mathématique développée dans le chapitre précédent et remaniée afin d'intégrer les concepts de POO présentés et de comprendre l'amélioration en termes de qualité du code produit

### Utilisation de la généricité via le concept de Template C++

- Qu'est ce que la généricité ?
- Application de la généricité à la définition de fonctions

### Référence

THIL1101

### Durée

5 jours / 35 heures

### Prix HT / stagiaire

2750€

### Objectifs pédagogiques

- Concevoir les principes fondamentaux de la conception objet et les pratiquer de façon effective en C++ au travers d'une application réalisée de façon itérative et incrémentale tout au long de la formation
- Appliquer les concepts de la programmation orientée objet
- Concevoir les nouveautés offertes par la dernière norme C++ 11 / 14

### Niveau requis

- Garantir posséder de bonnes connaissances d'un langage de programmation de type C, Java, C#, Python ou autre
- Garantir connaître les principes de la programmation orientée objet

### Public concerné

- Développeurs
- Chargés de développement d'applications informatiques
- Chefs de projets développement

### Formateur

Les formateurs intervenants pour Themanis sont qualifiés par notre Responsable Technique Olivier Astre pour les formations informatiques et bureautiques et par Didier Payen pour les formations management.

### Moyens pédagogiques et techniques

Salles de formation (accessibles et adaptables aux besoins des personnes en situation de handicap) équipée d'un ordinateur de dernière génération par stagiaire, réseau haut débit et vidéo-projection UHD

Documents supports de formation projetés  
Apports théoriques, étude de cas concrets et exercices

Mise à disposition en ligne de documents supports à la suite de la formation

### Dispositif de suivi de l'exécution de l'évaluation des résultats de la formation

Feuilles d'émargement (signature électronique privilégiée)

Evaluations formatives et des acquis sous forme de questions orales et/ou écrites (QCM)

- Application de la généricité à la définition de classes
- Template utilisant des types par défaut
- Utiliser plusieurs types de généricité
- TP : implémentation d'une classe générique permettant la manipulation sécurisée de tableaux

## Héritage et polymorphisme

- Qu'est ce que l'héritage
- Mise en oeuvre de l'héritage
- La redéfinition de méthode et le spécificateur override
- Liaison statique, liaison dynamique et le polymorphisme
- Le mot clé protected
- Héritage public, privé ou protégé
- L'héritage multiple en C++
- Finalisation de classes et de méthodes
- Assistance proposées par Eclipse/CDT
- TP : mise en oeuvre d'une hiérarchie de classes en utilisant les différents concepts étudiés
- Qu'est ce que l'héritage ?
- Traitement d'exceptions en C++
- Qu'est ce qu'une exception ?
- Comparatif avec le traitement d'erreurs en C
- Lever une exception dans le système
- Capturer, relayer et traiter une exception
- Coder une classe d'exception (entêtes et )
- Les principales classes d'exceptions proposées par la librairie standard
- TP : implémentation d'une classe d'exceptions, déclenchement d'exceptions et traitement de ces dernières

## Utilisation d'espaces de noms pour organiser votre code

- Qu'est qu'un espace de noms (namespace)
- Localiser un élément dans l'espace de noms global
- Le namespace std:: et la librairie standard

## Documenter vos codes

- Présentation de l'outil Doxygen
- Les différents marqueurs Doxygen
- TP : rédaction d'une documentation et génération via Doxygen

## La gestion des I/O en C++

- Les entêtes et
- Accès à des fichier textes (8 bits ou 16 bits)
- Correctement traiter les erreurs
- Accès à des fichiers binaires
- TP : manipulation de fichiers textes et de fichier binaires

## Mise en oeuvre de tests unitaires en C++

- Pourquoi faire des tests unitaires ?
- Présentation de CppUnit
- Intégration dans Eclipse/CDT
- TP : réalisation d'une batterie de jeux de tests unitaires

## Utilisation d'API de log

- Présentation de Log4cxx
- Les différents niveaux de logs
- Configuration du moteur Log4Cxx
- Outil graphique de visualisation de logs : chainsaw

- TP : production et analyse de logs