



Phirio

Langage C++ initiation

LC004

Durée: 4 jours

Public :

Tout développeur souhaitant apprendre le langage C++ pour le développement ou la maintenance de programmes.

Objectifs :

Connaître les principes de la programmation objet
Connaître et maîtriser la syntaxe du langage C++
Utiliser les outils de développement associés au langage C++

Connaissances préalables nécessaires :

Avoir suivi la formation "bases de la programmation" ou posséder des connaissances des techniques de base d'un langage de programmation structuré. Avoir pratiqué le langage C

Programme :

Concepts de base de la programmation objet.

Définitions : besoin, cadre d'utilisation, termes utilisés, organisation, historique.
Présentation des propriétés du modèle objet : uniformité, encapsulation, autonomie des objets, mécanisme d'abstraction, l'héritage, le polymorphisme
Les classes, les objets, les messages

Atelier : Ecriture de classes de base. Mise en évidence de la portée des accès lors d'un héritage.

Syntaxe C++. Structure d'un programme.

Architecture d'un programme écrit en C++; variables simples; affichage: cout et cin; fonctions; surcharges.
Structure d'un programme :
boucles, tests, branchements
Les variables
Les différents types de variables et attributs (tableaux, données statiques, constantes), règles de portées.
Autoréférence.
Allocation dynamique : new et delete.
Tableaux d'objets
Opérateurs : calcul, comparaison, binaires.

Atelier : Construction d'un tableau d'animaux 'chats' et 'chiens'.
Manipulations : tris, ajouts, suppressions



Phirio

Les classes en C++

Principe de la déclaration de classe
Les membres d'une classe, contrôle d'accès aux membres
Notions de constructeurs et destructeurs
L'instanciation d'une classe (déclarative, dynamique)

Atelier : Ecriture d'une classe 'chat' et d'une classe 'chien'.

Les membres spéciaux, méthodes constantes
Fonctions membres, fonctions inline, surcharges des membres, accès : public, private, protected.

Atelier : Travail sur les champs de 'chat' et 'chien' : visibilité, partage, ...

Fonctions amies.
Imbrication d'objets, dérivation, héritage.

Atelier : Ecriture d'une classe 'mammifère' et d'une hiérarchie associée.

Surcharge.

Surcharge d'opérateurs. Surcharge de && et >>.
Fonction-opérateur amie.
Le constructeur de copie, opérateur d'affectation, l'opérateur fonction

Atelier : surcharge des opérateurs sur la hiérarchie d'animaux. Surcharge de l'indexation. Ajout de fonctions-opérateurs amies.

Polymorphisme et héritage

Notion de classes dérivées
Le mécanisme d'héritage, héritage et instanciation
Le polymorphisme : conversion standards dans l'héritage, classes abstraites et méthodes virtuelles pures
L'héritage multiple.

Atelier : Ajout de fonctions abstraites dans la classe 'mammifère'

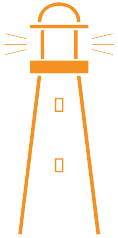
Atelier : Ajout d'une classe 'animal' globale. Mise en oeuvre de méthodes virtuelles pures.

Atelier : Mise en oeuvre du polymorphisme sur les classes 'chien' et 'chat'

Les templates.

Principe de fonctionnement,
programmation générique,
template de classe, de fonction
Alias de template et arguments

Atelier : Création d'une classe modèle d'animaux. Instanciation, utilisation.



Phirio

La STL

Présentation de la STL et des composants mis à disposition
Les conteneurs : vector, list, et, map
Mise en oeuvre sur des exemples simples
Les algorithmes : itérateurs, parcours et prédicats

Atelier : Création d'une liste chaînées d'animaux, parcours, recherche, Présentation de la STL et des composants mis à disposition

Les conteneurs : vector, list, et, map
Mise en oeuvre sur des exemples simples
Les algorithmes : itérateurs, parcours et prédicats

Atelier : Création d'une liste chaînées d'animaux, parcours, recherche, insertion, suppression

insertion, suppression

iostream

Description de la bibliothèque iostream : entrées/sorties
utilisation de la sortie pour un type prédéfini et pour un type utilisateur ;
entrée pour un type prédéfini et pour un type utilisateur ;
Mise en forme,
Manipulateur d'E/S standard

Atelier : Ecriture d'un système de sauvegarde de la hiérarchie d'animaux

Gestion des exceptions

Principe de traitement des erreurs
Traitement centralisé et polymorphisme
Exemples des exceptions standards

Atelier : ajout de cas d'erreurs sur la classe 'animal' et traitement, remontée des exceptions