

DÉVELOPPER EN C++

PUBLIC Salariés, demandeurs d'emploi, intermittents

PRÉ-REQUIS Bonnes notions en HTML/CSS et connaître les principes de bases de la conception d'un site web

DURÉE 35 heures (5 jours)

OBJECTIFS Développer les premières applications en C++, être capable de concevoir un programme stable, évolutif et réutilisable, intégrant les fonctions clés indispensables, devenir architectes de classes

FORMALISATION À L'ISSUE DE LA FORMATION : Attestation de fin de formation

INTRODUCTION À LA SYNTAXE DU LANGAGE C++

- Différences entre C et C++
- Données : définition, initialisation, types
- Expressions :
références, mécanismes de cast, conversions de types
- Notion d'opérateurs
- Références : arguments, valeurs de retour
- Fonctions Intégration de code C dans la programmation C++
- Type constants et espaces de nommage

CLASSES ET LES OBJETS C++

- Notion de syntaxe : champs, méthodes et constructeurs
- Fonctions
- Contrôle d'accès
- L'autoréférence
- Champs et méthodes statiques
- Création dynamique de tableaux d'objets
- Méthodes de conception des classes
- Méthodes et classes « Friend »
- Problématiques de gestion mémoire
- C++11 : constructeurs de copie, délégation, et déplacement

PROGRAMMATION ORIENTÉE OBJET

- Relation entre le C++ et la programmation objet
- Concepts des techniques orientées objet
- Méthodologies orientées objet
- Modèles et notation UML

EXCEPTIONS

- Concepts méthodologiques
- Aspects syntaxiques et génération d'exceptions

SURCHARGE DES OPÉRATEURS

- Concept de la surcharge
- Surcharge d'opérateurs : binaires /particuliers :
l'opérateur indice, fonction,
conversion/de gestion mémoire /'<<' et '>>'

DÉRIVATION ET HÉRITAGE

- Aspects syntaxiques : définition des classes dérivées, constructeurs
- Contrôle d'accès
- Fonctions virtuelles/polymorphisme
- Classes abstraites
- Interfaces
- Dérivation multiple
- Aspects sémantiques et méthodologiques

MODÈLES

- Modèle de classe : principes et mécanismes
- Surcharge des modèles et redéfinition de méthodes
- Modèle de fonction : principes et mécanismes
- Modèles et surcharge des opérateurs
- Modèles et les mécanismes de dérivation

I/O (INPUT/OUTPUT OU ENTRÉES/SORTIES)

- Principes I/O
- Principe des « streams »
- Classes d'entrée/sortie et leurs hiérarchies STL (Standard Template Library)
- Objectifs/principes, modèles et classes
- Conteneurs, itérateurs, boucle basée sur un intervalle avec C++ 11
- Types complémentaires, conteneurs, algorithmes, itérateurs, objets fonctions, adaptateurs
- Chaînes de caractères STL, classe « template basic_string »
- Conteneurs séquentiels et associatifs :
définition, rôle et critères de choix
- Allocateurs et gestion de la mémoire des conteneurs
- Vector, List, Set, Stack
- Itérateur/parcours d'un conteneur
- Groupes d'algorithmes STL :
mutants, non-mutants, tri/fusion, numériques
- Paramétrage d'algorithmes génériques via objets
« fonction »
- Manipulation de conteneurs
- « Adapteurs » et modification du comportement
d'un composant
- Traitements sur les flux
- RAll : pointeurs automatiques et classe auto_ptr
- Exceptions standard à la STL

GESTION DES OPÉRATEURS

- Opérateurs unaires, binaires - d'indirection - de
référencement - d'incrément/décément préfixés/
post-fixés
- Comparaison, affectation
- Surcharge de l'opérateur (), des opérateurs d'insertion
(<<) et d'extraction (>>)
- Foncteurs et surcharge de l'opérateur ()

Lieu de la formation : 146 rue Paradis - 13006 Marseille et en distanciel

Nombre de stagiaires (si en session) : 3 minimum - 12 maximum **Moyens pédagogiques :** 1 poste par stagiaire

1 Vidéoprojecteur - Plateforme E-learning **Modalités pédagogiques :** Alterne temps de présentation, d'échanges en groupe avec

l'intervenant et réflexion personnelle **Modalités d'évaluation :** Cas pratique en continu avec évaluation par le formateur

DÉVELOPPER EN C++ - SUITE

CONVERSION ET RTTI

- Opérateurs de conversion de casting /const_cast, static_cast, reinterpret_cast/ typeid/exceptions liées
- Constructions implicites, le mot-clé explicit
- Conversion dynamique/Runtime Type Information
- Classe type_info
- « Downcasting », opérateur dynamic_cast

GÉNÉRICITÉ

- Patrons de classe
- Généricité et préprocesseur
- Fonction générique
- Classe générique
- Composition générique
- Généralisation générique
- Spécialisation partielle/totale
- Méta-programmation

PROGRESSION VERS C++11

- Délégation de constructeurs
- Énumérations « type safe »
- Mot-clé auto et boucle sur intervalle
- Référence rvalue et impact sur la forme normale des classes C++
- Lambda-expressions
- Nouveaux conteneurs : array, forward_list, unordered_set, unordered_map
- Classe tuple
- Smart pointers : shared_ptr, weak_ptr, unique_ptr
- Nouveautés foncteurs et binders
- Gestion des threads
- Expressions régulières

BOOST

- Pointer Container Library
- Structures de données boost::any, boost::variant
- Programmation événementielle – connexions/signaux
- Gestion des processus
- Mécanismes de communication interprocessus
- Mémoire partagée
- Utilisation avancée de l'héritage
- Héritage versus embarquement /privé /protégé/ multiple, gestion de collisions de membre/ en diamant/ virtuel et dynamic_cast
- Exportation de membres cachés via la Clause Using
- Principes de conception : substitution de Liskov, principe d'ouverture/fermeture, inversion des dépendances
- Règles d'implémentation des interfaces en C++

Lieu de la formation : 146 rue Paradis - 13006 Marseille et en distanciel

Nombre de stagiaires (si en session) : 3 minimum - 12 maximum **Moyens pédagogiques :** 1 poste par stagiaire

1 Vidéoprojecteur - Plateforme E-learning **Modalités pédagogiques :** Alterne temps de présentation, d'échanges en groupe avec l'intervenant et réflexion personnelle **Modalités d'évaluation :** Cas pratique en continu avec évaluation par le formateur