

**PLAN DE FORMATION**  
**PROGRAMMATION LANGAGE C++**  
**Initiation**

Public visé : Toute personne ayant déjà des notions de programmation dans un langage quelconque.

Objectifs : Initiation au langage C++ et apprentissage des bases fondamentales.

Moyens pédagogiques : Formation dirigée par un professionnel des systèmes d'informations et de la programmation orientée objet en C++.

Méthodes d'animation actives et pratiques. Elles s'appuient sur des apports théoriques et méthodiques (exercices, études de cas).

Méthodes d'évaluation des acquis : Après chaque session, une évaluation des acquis précédents sera envisagée, suivra un mini projet en fin de formation.

Contenu de la formation :

## **Chapitre1**

### 1. Première approche du C/C++

- 1.1. Les commentaires en C++
- 1.2. Les types prédéfinis du C/C++
- 1.3. Notation des valeurs
- 1.4. La définition des variables
- 1.5. Instructions et opérations
- 1.6. Les fonctions
- 1.7. La fonction main
- 1.8. Les fonctions d'entrée / sortie de base
- 1.9. Exemple de programme complet

## **Chapitre 2**

### 2. Les structures de contrôle

- 2.1. La structure conditionnelle if
- 2.2. La boucle for
- 2.3. Le while
- 2.4. Le do
- 2.5. Le branchement conditionnel
- 2.6. Le saut
- 2.7. Les commandes de rupture de séquence

.../...

.../...

### **Chapitre 3**

#### 3. Types avancés et classes de stockage

- 3.1. Structures de données et types complexes
- 3.2. Les classes de stockage

### **Chapitre 4**

#### 4. Les pointeurs et références

- 4.1. Notion d'adresse
- 4.2. Notion de pointeur
- 4.3. Déréférencement, indirection
- 4.4. Notion de référence
- 4.5. Lien entre les pointeurs et les références
- 4.6. Passage de paramètres par variable ou par valeur
- 4.7. Références et pointeurs constants et volatiles
- 4.8. Arithmétique des pointeurs
- 4.9. Utilisation des pointeurs avec les tableaux
- 4.10. Les chaînes de caractères : pointeurs et tableaux à la fois !
- 4.11. Allocation dynamique de mémoire
- 4.12. Pointeurs et références de fonctions
- 4.13. Paramètres de la fonction main - ligne de commande
- 4.14. DANGER

### **Chapitre 5**

#### 5. Le préprocesseur C

- 5.1. Définition
- 5.2. Les commandes du préprocesseur
- 5.3. Les macros
- 5.4. Manipulation de chaînes de caractères dans les macros
- 5.5. Les trigraphes

### **Chapitre 6**

#### 6. Modularité des programmes et génération des binaires

- 6.1. Pourquoi faire une programmation modulaire ?
- 6.2. Les différentes phases du processus de génération des exécutables
- 6.3. Compilation séparée en C/C++
- 6.4. Syntaxe des outils de compilation
- 6.5. Problèmes syntaxiques relatifs à la compilation séparée

### **Chapitre 7**

#### 7. Comment faire du code illisible ?

### **Chapitre 8**

#### 8. C++ : la couche objet

- 8.1. Généralités
- 8.2. Extension de la notion de type du C

.../...

.../fin

- 8.3. Déclaration de classes en C++
- 8.4. Encapsulation des données
- 8.5. Héritage
- 8.6. Classes virtuelles
- 8.7. Fonctions et classes amies
- 8.8. Constructeurs et destructeurs
- 8.9. Pointeur this
- 8.10. Données et fonctions membres statiques
- 8.11. Surchage des opérateurs
- 8.12. Des entrées - sorties simplifiées
- 8.13. Méthodes virtuelles
- 8.14. Dérivation
- 8.15. Méthodes virtuelles pures - Classes abstraites
- 8.16. Pointeurs sur les membres d'une classe

## **Chapitre9**

### 9. Les exceptions en C++

- 9.1. Lancement et récupération d'une exception
- 9.2. Remontée des exceptions
- 9.3. Liste des exceptions autorisées pour une fonction
- 9.4. Hiérarchie des exceptions
- 9.5. Exceptions dans les constructeurs

## **Chapitre10**

### 10. Identification dynamique des types

- 10.1. Identification dynamique des types
- 10.2. Transtypes C++

## **Chapitre11**

### 11. Les espaces de nommage

- 11.1. Définition des espaces de nommage
- 11.2. Déclaration using
- 11.3. Directive using

## **Chapitre12**

### 12. Les template

- 12.1. Généralités
- 12.2. Déclaration des paramètres template
- 12.3. Fonctions et classes template
- 12.4. Instanciation des template
- 12.5. Spécialisation des template
- 12.6. Mot-clé typename
- 12.7. Fonctions exportées

## **Chapitre13**

### La bibliothèque standard C++