

Langage C++ Programmation

PLAN DE COURS

Objectifs : Acquérir les principes de base de la programmation Objet (polymorphisme, héritage, encapsulation), maîtriser le langage C++, réaliser des applications et utiliser les "best practices" de tests de qualité en C++	
Prérequis : Avoir suivi la formation « Langage C – Introduction à la programmation » ou connaissances équivalentes	
Public : Développeurs, concepteurs.	
Niveau : Intermédiaire	Durée standard préconisée : 30 h
Pédagogie : alternance d'apports théoriques et nombreux exercices de mise en pratique	
Moyens pédagogiques : un ordinateur multimédia par apprenant, ordinateur et vidéoprojecteur pour l'animateur	Évaluation des acquis : Mise en pratique à l'aide d'exercices en autonomie puis corrigés individuellement et collectivement
Suivi après formation : <ul style="list-style-type: none"> • Certificat de stage • Un ouvrage de référence (remis en formation) 	

Détail des objectifs

Appréhender la programmation Objet

- Connaître l'historique des langages et leur évolution
- Définir les critères de qualité dans un développement logiciel
- Appréhender les langages orientés Objet dans leur généralité.....

Appréhender les concepts Objet

- Gérer classes et objets
- Gérer l'encapsulation (visibilité public-private-protected, namespaces)
- Gérer les membres et méthodes de classes (static)
- Définir l'héritage simple.....
- Définir l'héritage multiple (classe de base virtuelle).....
- Gérer le polymorphisme (virtuel) et les classes abstraites (virtuel pures)
- Gérer les interfaces.....

Appréhender la syntaxe C++.....

- Utiliser les fonctions « Friends »
- Utiliser les fonctions « Inline »
- Utiliser le paramètre caché this
- Gérer le point d'entrée main avec arguments.....
- Définir le type références.....
- Gérer les classes imbriquées (Inner class)
- Inclure des headers et références multiples.....
- Utiliser le qualificateur const
- Utiliser les bibliothèques standard d'E/S.....
- Intégrer la gestion dynamique de la mémoire.....
- Utiliser les destructeurs virtuels
- Définir le typage dynamique avec RTTI.....

Traiter les exceptions

- Traiter des erreurs dans les programmes
- Traiter des erreurs en C++
- Traiter des exceptions imbriquées.....
- Gérer les classes d'exception applicatives

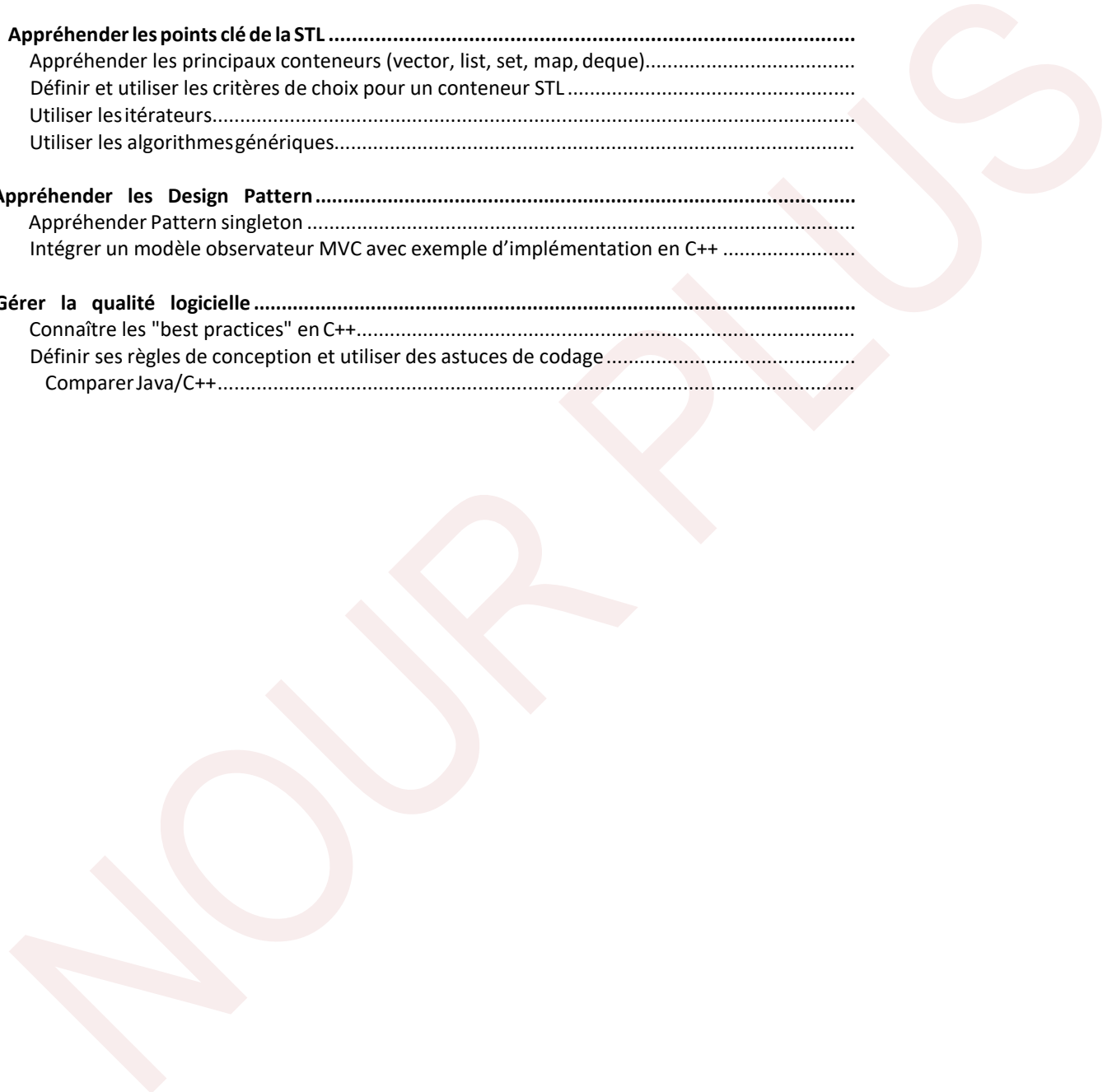
Gérer la surcharge.....

Gérer la surcharge des fonctions
Gérer la surcharge d'opérateurs.....
Gérer la surcharge de l'opérateur new (avec nothrow) et set-new_handler
Gérer les patrons- « TEMPLATES ».....
Définir des patrons, syntaxe et instanciation.....
Gérer des patrons de fonctions.....
Gérer des patrons de classes (exemple avec la classe smart pointer)

Appréhender les points clé de la STL
Appréhender les principaux conteneurs (vector, list, set, map, deque).....
Définir et utiliser les critères de choix pour un conteneur STL

Appréhender les Design Pattern.....
Appréhender Pattern singleton

Gérer la qualité logicielle
Connaître les "best practices" en C+.....
Définir ses règles de conception et utiliser des astuces de codage



NOUR PLUS