

POE Développeurs C++ QT

57 j (399 heures)

Ref : POE-C++

Public

Bac +3 avec expérience significative en Informatique
Bac +5 à Bac +8 Scientifique ou Informatique

Pré-requis

Réussite de nos tests de recrutement
Bon relationnel, ouvert, curieux, communicant
Niveau correct en Anglais

Moyens pédagogiques

Questionnaire d'évaluation de la satisfaction en fin de stage
Feuille de présence émargée par demi-journée par les stagiaires et le formateur
Attestation de fin de formation

Modalités de suivi et d'évaluation

Feuille de présence émargée par demi-journée par les stagiaires et le formateur
Exercices de mise en pratique ou quiz de connaissances tout au long de la formation permettant de mesurer la progression des stagiaires
Questionnaire d'évaluation de la satisfaction en fin de stage
Auto-évaluation des acquis de la formation par les stagiaires
Attestation de fin de formation

Le développeur C/C++ Qt est capable d'évoluer à la fois sur la partie front et back d'une application. Il peut développer des interfaces utilisateurs modernes et adaptables à tout type de support (téléphone, tablette, ordinateur). L'utilisation du langage C/C++ permet d'adresser toutes les contraintes matérielles et besoins en performance. L'utilisation du standard évolutif C++ allie les atouts d'une programmation performante de bas niveau (gestion de la mémoire, multithreading) avec celle de haut niveau (maintenabilité et réutilisabilité). Le développeur C/C++ applique ainsi les principes des programmations orientée objets, fonctionnelle et générique à une application performante qui gagne en qualité. Enfin le développeur C/C++ Qt maîtrise les outils DevOps pour s'intégrer dans une équipe de projet et une chaîne d'intégration continue de qualité.

Objectifs

Analyser les besoins fonctionnels et techniques
Rédiger un cahier des charges

- Maitriser le langage SQL
- Modéliser les caractéristiques d'une application
- Développer des applications avec une interface utilisateur moderne et fluide
- Développer des composants robustes et efficaces
- Tester ses programmes et applications, apporter des solutions
- Travailler en équipe dans une chaîne d'intégration continue de qualité, en mode Agile

Programme détaillé

TEAM BOOSTER

- Créer une cohésion de groupe
- Travailler son savoir-être en équipe
- Se connaître : ses points forts et ses axes de progrès
- Savoir se présenter
- Connaître les autres et s'enrichir de la diversité : profils de personnalité, cursus de formation, expérience
- Assimiler la puissance de la notion d'intelligence collective

LABS ALGORITHMIQUE

- Création d'algorithmes simples
- Comprendre leurs coûts
- Description et Utilisation des différentes boucles
- Calcul de vitesse d'un algorithme
- Les structures de données

LANGAGE C

- Structure générale d'un programme
- Les différents opérateurs
- Les structures de contrôle
- L'allocation dynamique de la mémoire
- Les entrées / sorties
- Les paramètres de la ligne de commande
- Structures complexes et allocation dynamique

POO ET ARCHITECTURE LOGICIELLE

- De l'analyse à l'objet
- Pourquoi l'objet ?
- Les bases et concepts de l'objet
- Conception d'une architecture informatique à base d'objets
- Les outils de développement objet

OVERVIEW DES METIERS ESN

Définition ESN
Les métiers du Développement PLM
Les conditions d'exercices
L'évolution en ESN

C++

La fonction main et les E/S standard
Variables, constantes et références
Définition et appels de fonctions
Décisions, boucles et logique
Tableaux, pointeurs et chaînes
Encapsulation des types de données de niveau supérieur
Déclaration, modification et accès aux objets
Surcharge des opérateurs et des fonctions
Séparation des interfaces et des mises en œuvre
Dérivation de nouvelles classes des classes existantes
Utilisation de fonctions polymorphiques

BASES DE DONNEES

Le relationnel
L'algèbre relationnelle et la logique du SQL
L'interrogation des données
Les jointures
Les sous-interrogations
Mise à jour des données
Sécurité et transactions
Traitement des Requêtes SQL
Les outils de mesure de performance
Mécanismes automatiques de réglage de performances
Optimisation du schéma relationnel
Optimisation du serveur
Optimisation des requêtes SQL
Parallélisme d'exécution des requêtes

QT & QML

Présentation du langage Qt
Les différents outils de Qt
Les éléments importants d'un projet
Les différents composants graphiques
La gestion des événements
Les fichiers et les types de données

- Les QWidgets personnalisés et le dessin vectoriel
- Le model vue contrôleur avec Qt
- Le réseau, la communication
- Les threads
- Internationalisation
- QML
- Les types de base QML
- QT Quick
- Communication QML / C++

MISE EN PROJET

- Présentation du projet final
- Mise en équipe !
- Réception du projet et des spécifications
- Analyse des demandes
- Savoir proposer des évolutions ou améliorations
- Rédaction d'une conception UML
- Identification des difficultés
- Itérer pour améliorer le produit
- Construction d'une POC (Preuve de Concept)
- Création des tests unitaires
- Calcul du cout du projet
- Respecter les délais
- Estimation du reste à faire et des améliorations

AGILITE SCRUM / CYCLE EN V

- Cycle en V
- AMOE
- AMOA
- Exemple de procédure
- Agilité
- Programmation XP
- Pair Programming
- Test driven Development
- Méthodologie Scrum
- Gestion du temps
- Définition d'un projet
- Responsabilité du chef de projet
- Réutilisation d'un projet dans un autre

UML + DESIGN PATTERN

- De l'analyse à l'objet
- Pourquoi l'objet ?

- Les bases et concepts de l'objet
- Conception d'une architecture informatique à base d'objets
- Les outils de développement objet
- Utilisation des objets dans les langages architectures principaux
- UML et le développement logiciel
- Diagrammes UML
- Cas d'utilisation (use cases)
- Les Design Patterns du GoF et autres Patterns
- Patterns et architecture

GIT

- Installation et configuration
- Principes de git
- Guide pratique de git
- Erreurs, imprévus : se sortir d'un mauvais pas
- Git en détail
- Les sous-modules
- Git en équipe

ADMINISTRATION LINUX

- Installation et systèmes de fichiers
- Commandes de bases (Manipulation des fichiers et des dossiers)
- Installation des paquets
- Le réseau (configuration ip, dns...)
- Processus et services
- Les traces du système (gestion des logs système et applicatif)
- Administrer à distance (SSH)
- Introduction au Shell

C++ AVANCE

- Programmation fonctionnelle avec C++
- Utilisation avancée de la généricité
- Multithreading

TESTER SON CODE

- Introduction aux tests des applications
- Tests unitaires
- Tests fonctionnels et recettes
- Tests de montée en charge
- Optimisation et profiling des applications
- Introduction au TDD

DEVOPS

Jenkins

Automatisation des build

Couverture de code

Analyse statique

Docker

Commandes de base
