

Préparation à la certification "70-483 - Programming in C#"

Référence : 19003-161222-15-PRO

Durée : 1 jour soit 07 heures

Filière : Développement

Stage disponible en mode « In Class »

Population visée :

Ouvrier – Employé – Employé qualifié – Cadre – Cadre supérieur

Public concerné :

Développeurs, analystes programmeurs, chefs de projets.

Cette formation de préparation au passage de l'examen de certification " 70-483 - Programming in C# ", permet de réviser les points essentiels sur lesquels les candidats pourront être interrogés. Le plan de cours détaille l'ensemble des points de l'examen. Avant cette formation, les candidats auront déjà réalisé un travail personnel important en vue de se présenter à l'examen. Lors de la formation de préparation, le formateur parcourt les différents points de l'examen et s'attarde sur ceux qui peuvent poser problème aux candidats ou ceux pour lesquels il peut y avoir des pièges à éviter ou des cas particuliers.

PRÉ-REQUIS

- Il est fortement recommandé d'avoir au minimum 6 mois d'expérience pratique en développement C#.
- Avoir suivi la formation " - Les fondamentaux du développement .NET avec le langage C# 6 sous Visual Studio 2015 " ne constitue pas un élément suffisant pour se présenter au passage de l'examen.
- Cette formation ne dispense pas les candidats d'un travail personnel important en vue de se présenter au passage de l'examen.

OBJECTIFS PEDAGOGIQUES

A l'issue de cette formation, l'apprenant sera capable de :

- Vous présenter au passage de l'examen " 70-483 - Programming in C# ".

OBJECTIFS OPERATIONNELS ET CONTENU DE LA FORMATION

Gestion du flux de programme

- Mise en œuvre du multithreading et du traitement asynchrone
 - ✓ Utilisation de la bibliothèque Task Parallel (ParallelFor, Plinq, Tasks)
 - ✓ Création de tâches de continuation
 - ✓ Génération dynamique des threads à l'aide de ThreadPool
 - ✓ Déblocage de l'interface utilisateur
 - ✓ Utilisation des mots clés async et await
 - ✓ Gestion des données à l'aide des collections simultanées
- Gestion du multithreading
 - ✓ Synchronisation des ressources
 - ✓ Mise en œuvre du verrouillage
 - ✓ Annulation d'une tâche à long terme
 - ✓ Mise en œuvre de méthodes thread-safe pour gérer les conditions de concurrence critique
- Mise en œuvre du flux de programme
 - ✓ Itération à travers les éléments de collecte et de tableau
 - ✓ Programmation des décisions à l'aide d'instructions switch, si/alors et d'opérateurs
 - ✓ Evaluation des expressions

- Création et mise en œuvre des événements et des rappels
 - ✓ Création de questionnaires d'événements
 - ✓ Abonnement aux événements et désabonnement
 - ✓ Utilisation de types délégués intégrés pour créer des événements
 - ✓ Création de délégués
 - ✓ Expressions lambda
 - ✓ Méthodes anonymes
- Mise en œuvre de la gestion des exceptions
 - ✓ Gestion des types d'exception (exceptions SQL, exceptions réseau, exceptions de communication, exceptions de délai d'expiration du réseau)
 - ✓ Exceptions de type catch par rapport à des exceptions de base
 - ✓ Mise en œuvre des blocs try-catch-finally
 - ✓ Levée d'exceptions
 - ✓ Détermination du moment auquel relever par rapport à lever
 - ✓ Création d'exceptions personnalisées

Création et utilisation des types

- Création de types
 - ✓ Création de types de valeur (structs, enum), types de référence, types génériques, constructeurs, variables statiques, méthodes, classes, méthodes d'extension, paramètres en option et nommés et propriétés indexées
 - ✓ Création de méthodes surchargées et remplacées
- Consommation de types
 - ✓ Utilisation des fonctions Box ou Unbox pour convertir les types de valeur
 - ✓ Types de conversion
 - ✓ Gestion des types dynamiques
 - ✓ Garantie de l'interopérabilité avec du code non géré, par exemple, mot clé dynamique
- Application de l'encapsulation
 - ✓ Application de l'encapsulation à l'aide de propriétés, d'accessesurs (public, privé, protégé) et de la mise en œuvre de l'interface explicite
 - ✓ Création et mise en œuvre d'une hiérarchie de classe
 - ✓ Conception et mise en œuvre d'une interface
 - ✓ Héritage d'une classe de base
 - ✓ Création et mise en œuvre de classes en fonction des interfaces IComparable, IEnumerable, IDisposable et IUnknown
- Recherche, exécution et création de types lors du runtime à l'aide de la réflexion
 - ✓ Création et application d'attributs
 - ✓ Lecture d'attributs
 - ✓ Génération de code lors du runtime à l'aide des expressions CodeDom et lambda
 - ✓ Utilisation de types à partir de l'espace de noms System.Reflection (Assembly, PropertyInfo, MethodInfo, Type)
- Gestion du cycle de vie d'un objet
 - ✓ Gestion des ressources non gérées
 - ✓ Mise en œuvre de l'interface IDisposable, y compris l'interaction avec finalisation
 - ✓ Gestion de l'interface IDisposable à l'aide de l'instruction Using
 - ✓ Gestion des opérations de finalisation et garbage collection
- Manipulation des chaînes
 - ✓ Manipulation des chaînes à l'aide des classes StringBuilder, StringWriter et StringReader
 - ✓ Chaînes de recherche
 - ✓ Énumération des méthodes de chaîne
 - ✓ Formatage des chaînes

Débogage d'applications et mise en œuvre de la sécurité

- Validation des entrées de l'application
 - ✓ Validation des données JSON
 - ✓ Types de collecte de données

- ✓ Gestion de l'intégrité des données
- ✓ Evaluation d'une expression régulière pour valider le format d'entrée
- ✓ Utilisation de fonctions intégrées pour valider le type de données et le contenu hors de portée
- ✓ Ecriture d'expressions régulières
- Réalisation d'un chiffrement symétrique et asymétrique
 - ✓ Choix d'un algorithme de chiffrement approprié
 - ✓ Gestion et création des certificats
 - ✓ Mise en œuvre de la gestion des clés
 - ✓ Mise en œuvre de l'espace de noms System.Security
 - ✓ Données de hachage
 - ✓ Chiffrement des flux
- Gestion des assemblages
 - ✓ Contrôle des versions des assemblages
 - ✓ Signature des assemblages à l'aide de noms forts
 - ✓ Mise en œuvre de l'hébergement côte à côte
 - ✓ Mise en Global Assembly Cache
 - ✓ Création d'un assemblage WinMD
- Débogage d'une application
 - ✓ Création et gestion des directives du compilateur
 - ✓ Choix d'un type de build approprié
 - ✓ Gestion de la programmation des fichiers de la base de données et des symboles
- Mise en œuvre du diagnostic dans une application
 - ✓ Mise en œuvre de la journalisation et du traçage
 - ✓ Profilage des applications
 - ✓ Création et analyse des compteurs de performances
 - ✓ Ecriture dans le journal des événements

Mise en œuvre de l'accès aux données

- Exécution des opérations d'E/S
 - ✓ Lecture et écriture des fichiers et des flux
 - ✓ Lecture et écriture à partir du réseau à l'aide de classes dans l'espace de noms System.Net
 - ✓ Mise en œuvre des opérations d'E/S asynchrones
- Utilisation des données
 - ✓ Récupération des données à partir d'une base de données
 - ✓ Mise à jour des données dans une base de données
 - ✓ Consommation des données JSON et XML
 - ✓ Récupération des données à l'aide des services Web
- Interrogation et manipulation des données et des objets à l'aide de LINQ
 - ✓ Interrogation des données à l'aide d'opérateurs (projection, rejoindre, grouper, prendre, sauter, agrégat)
 - ✓ Création de requêtes LINQ basées sur une méthode
 - ✓ Interrogation des données à l'aide d'une syntaxe de compréhension de requête
 - ✓ Sélection des données à l'aide de types anonymes
 - ✓ Forçage de l'exécution d'une requête
 - ✓ Lecture, filtrage, création et modification des structures de données à l'aide de LINQ to XML
- Sérialisation et désérialisation des données
 - ✓ Sérialisation et désérialisation des données à l'aide de la sérialisation binaire, sérialisation personnalisée, sérialiseur XML, sérialiseur JSON et sérialiseur de contrat de données
- Stockage des données dans les collections et extraction des données de celles-ci
 - ✓ Stockage et récupération des données à l'aide de dictionnaires, tableaux, listes, ensembles et files d'attente
 - ✓ Choix d'un type de collection
 - ✓ Initialisation d'une collection
 - ✓ Ajout d'éléments à une collection et retrait d'éléments de celle-ci
 - ✓ Utilisation des collections typées par rapport aux collections non typées
 - ✓ Mise en œuvre de collections personnalisées
 - ✓ Mise en œuvre d'interfaces de collection

Méthodes et moyens :

- Explications théoriques suivies de pratiques guidées puis mises en autonomie
- 1 vidéoprojecteur par salle
- Stage en mode « In Class » 2 téléviseurs et une caméra HD par salle
- 1 ordinateur par stagiaire

Méthodes d'évaluation des acquis :

- Exercices de synthèse et d'évaluation
- Evaluation de fin de stage

Profil formateur :

Nos formateurs sont certifiés à l'issue d'un parcours organisé par nos soins. Ils bénéficient d'un suivi de compétences aussi bien au niveau technique que pédagogique.

Support stagiaire :

- Support papier ou électronique (dématérialisé)
- Les exercices d'accompagnement peuvent être récupérés sur clef USB