

Plan de cours N° : 1188

Durée : 3 jours (21h)

PARTICIPANTS / PRE-REQUIS

Développeurs informatiques en .Net / C#

Maîtriser la programmation orientée objet avec le langage C# (version 7.x idéalement)

OBJECTIFS PEDAGOGIQUES

Appliquer les tests unitaires en C#. Evaluer les nouveautés de C# 7.x. Evaluer les fonctionnalités avancées de C# 8. Différencier les notions avancées sur les types. Etablir la gestion de la mémoire en C#. Contrôler l'asynchronisme en C#. Employer les capacités de LINQ. Apprécier le Garbage Collector. Appliquer les patterns utilisés en .NET Core.

MOYENS PEDAGOGIQUES

Réflexion de groupe et apports théoriques du formateur

Travail d'échange avec les participants sous forme de réunion-discussion

Utilisation de cas concrets issus de l'expérience professionnelle

Validation des acquis par des exercices de synthèse

Alternance entre apports théoriques et exercices pratiques (en moyenne 30 et 70%)

Remise d'un support de cours.

Assistance post-formation d'une durée de 1 an sur le contenu de la formation via notre adresse mail dédiée formateurs@atp-formation.com

MOYENS PERMETTANT LE SUIVI DE L'EXECUTION ET DES RESULTATS

Feuille de présence signée en demi-journée,

Evaluation des acquis tout au long de la formation,

Questionnaire de satisfaction,

Attestation de stage à chaque apprenant,

Positionnement préalable oral ou écrit,

Evaluation formative tout au long de la formation,

Evaluation sommative faite par le formateur ou à l'aide des certifications disponibles.

MOYENS TECHNIQUES EN PRESENTIEL

Accueil des stagiaires dans une salle dédiée à la formation, équipée d'ordinateurs, d'un vidéo projecteur d'un tableau blanc.

MOYENS TECHNIQUES DES CLASSES A DISTANCE

A l'aide d'un logiciel comme Teams, Zoom etc... un micro et éventuellement une caméra pour l'apprenant, suivez une formation en temps réel et entièrement à distance. Lors de la classe en ligne, les apprenants interagissent et communiquent entre eux et avec le formateur.

Les formations en distanciel sont organisées en Inter-Entreprise comme en Intra-Entreprise. L'accès à l'environnement d'apprentissage (support de cours, labs) ainsi qu'aux preuves de suivi et d'assiduité (émargement, évaluation) est assuré.

Les participants recevront une convocation avec lien de connexion

Pour toute question avant et pendant le parcours, une assistance technique et pédagogique est à disposition par mail et par téléphone auprès de notre équipe par téléphone au 04.76.41.14.20 ou par mail à contact@atp-formation.com

ORGANISATION

Les cours ont lieu de 9h00-12h30 13h30-17h00.

PROFIL FORMATEUR

Nos formateurs sont des experts dans leurs domaines d'intervention

Leur expérience de terrain et leurs qualités pédagogiques constituent un gage de qualité.

ACCESSIBILITE

Les personnes atteintes de handicap souhaitant suivre cette formation sont invitées à nous contacter directement, afin d'étudier ensemble les possibilités de suivre la formation.

MISE A JOUR

28/12/2023

Plan de cours N° : 1188

Durée : 3 jours (21h)

Les tests unitaires en C#

- Introduction aux tests unitaires
- Différents concepts de test
- Présentation des outils .Net pour les tests
- Simulation avec les Mock
- Le BDD (Behavior Driven Development) pour faire rentrer les opérationnels dans les tests
- Travaux pratiques
Ecrire un test unitaire en C# et le faire muter en BDD

C# 7.x : Nouveautés et fonctionnalités avancées

- Rappels de l'existant pré-C# 7
- Fonctions locales
Avantages et performances
- Tuples
- Déconstructeurs
- Pattern matching
- Out & discard
- Améliorations sur l'écriture des numériques
- Expression fléchée
- Expression throw
- Retour de fonction par référence
- Travaux pratiques
Plusieurs exercices pour mettre en pratique les nouveautés de C# 7

C# 8 : Nouveautés et fonctionnalités avancées

- Références sur les types valeurs
- Type référence nullable
- La portée private protected
- Code non sécurisé
- Indices et plages
- Implémentation par défaut des interfaces
- Travaux pratiques
Plusieurs exercices pour mettre en pratique les nouveautés de C# 8

Notions avancées sur les types

- Generics et contraintes de generics
- Réflexion simple
Création et introspection des types
- Réflexion avancée
Gestion des generics (méthodes, types)
- Types anonymes et dynamic
- Durée de vie avec IDisposable et les destructeurs
- Attributs
- Travaux pratiques
Créer dynamiquement des objets au runtime pour les types de questions

La mémoire en C#

- Introduction
- Comprendre les concepts de pile & tas
- Différences entre types valeur & types référence
Impacts et analyses
- La classe Span<T>
- Stackalloc
- Travaux pratiques
Plusieurs exercices pour comprendre les allocations mémoire et l'impact du code

Asynchronisme

- Reprise des concepts de l'asynchronisme
- La classe Thread
- La classe Parallel
- Définition et approfondissement sur le Task
- Le scheduler et créer son propre scheduler
- Async/await : détails et fonctionnement de ce qui est compilé réellement
- Gestion de l'annulation, des exceptions et de la continuation
- Les nouveautés C# 8 sur l'asynchronisme
- Travaux pratiques
Exercice de code pour montrer les différents pattern asynchronisme.
Exercice complet permettant de mettre en pratique tout le contenu

Plan de cours N° : 1188

Durée : 3 jours (21h)

LINQ

- Comment fonctionne LINQ ?
- Les différents opérateurs existants
- Comparaison de performance avec analyse IL
- Créer ses propres méthodes LINQ
- L'arbre d'expression
- Travaux pratiques
 - Création de son propre provider LINQ
 - Découverte de la classe Expression

Le garbage collector

- Rappel sur le rôle du garbage collector
- Algorithme du garbage collector
- Différences entre les ressources managées et non managées
- Comment soulager la charge du garbage collector
- Le pattern IDisposable
- Travaux pratiques
 - Créer une application et benchmarker les différences entre le passage du garbage collector

Patterns utilisés en .Net Core

- Présentation rapide de l'architecture d'un projet ASP.Net Core
- IoC et principes solides associés
 - Pourquoi, c'est utile
- Pattern Options pour gérer la configuration
- Fonctionnement du pipeline ASP.Net Core

