Introduction à la programmation avec C#

Objectifs	Ce stage vous permettra de comprendre les fondements de la programmation et de l'algorithmique. Vous acquérez des bases en programmation qui vous permettront d'aborder n'importe quel langage dans les meilleures conditions. Tous les aspects essentiels seront vus : les modèles de programmation, les éléments de lexique et de syntaxe, les outils, l'organisation du code, l'accès aux bases de données et les tests.
Participants	Toute personne devant apprendre à programmer.
Prérequis	Aucune connaissance particulière.
Moyens pédagogiques	1 poste par participant - 1 Vidéo projecteur - Support de cours fourni à chaque participant – Formation présentielle
Durée	3 jours

Code : MSC#-INTRO

Programme.

Un programme

Qu'est-ce qu'un programme ?
Qu'est-ce qu'un langage ? Les différents paradigmes.
Quel langage pour quelle application ?
Les compilateurs. Les exécutables.
Les responsabilités d'un programmeur.
Travaux pratiques
Présentation de différents langages (Java, C#, Visual Basic, C, C++).

Nécessité d'un algorithme

Qu'est-ce qu'un algorithme ?
Les besoins auxquels répond un algorithme.
Le concept de pseudo-langage.
Travaux pratiques
Ecriture d'un premier algorithme en pseudo-langage.

Genèse d'un premier programme

Ecriture d'un programme simple : syntaxe et instructions.
Compilation et exécution du programme.
Qu'est-ce qu'une librairie ? Son rôle, son usage.
Travaux pratiques
Découverte de l'environnement de développement et d'exécution. Ecriture, compilation et exécution d'un premier programme.

Règles de programmation

Convention de nommage.
Convention syntaxique.
Utilisation des commentaires. Pourquoi commenter les développements ?
Améliorer la lisibilité des programmes : indentation du code, découpage du code...

Les variables

Qu'est-ce qu'une variable ?
Pourquoi typer une variable ?
Les types primitifs : entiers, chaînes de caractères, nombres réels, autres.
Déclaration, définition et initialisation d'une variable.
Les constantes.
Saisie, affichage, affectation, conversion de type.
Organiser ses données sous forme de tableaux.
Les types évolués : enregistrement, matrice, arbre.
Travaux pratiques
Ecriture de plusieurs programmes simples manipulant les variables.

Opérateurs et expressions

Les différents opérateurs (multiplicatif, additif, comparaison, égalité, logique, affectation).
Combinaison d'opérateurs.
Expression booléenne.
Travaux pratiques
Manipulation des opérateurs et des expressions booléennes.

Les structures de contrôle

Les sélections alternatives (si, si-alors-sinon, sélection cas).
Les blocs d'instructions (notion de Début... Fin).
Les boucles itératives (tant que-répéter, répéter-jusqu'à, pour-de- à).
Imbrication des instructions.
Les commentaires.
Travaux pratiques
Utilisation des structures de contrôle pour implémenter un algorithme.

Les procédures et les fonctions

Définitions : procédure, fonction.
Pourquoi sont-elles incontournables en programmation (réutilisabilité, lisibilité...) ?
Le passage de paramètres.
Le code retour d'une fonction.
Sensibilisation aux limites du passage de la valeur d'une variable.
Notion de passage par adresse.
Appel de fonctions.

Introduction à la programmation objet

Les concepts associés à la programmation objet : classe, attribut, méthode, argument.
La modélisation objet à partir des exigences fonctionnelles : introduction aux bonnes pratiques d'organisation de conception et d'organisation d'un programme.
Travaux pratiques
Illustration des concepts objets.

L'accès aux bases de données

Organisation et stockage des données.
Les traitements de base (connexion, requêtes, récupération des données).
Application cliente et serveur de données.
Affichage et manipulation des données dans l'application cliente.
Travaux pratiques
Création d'un formulaire de recherche d'informations dans une base de données.

Maintenance, débogage et test des programmes

Savoir lire et interpréter les différents messages d'erreurs.
Utiliser un débogueur : exécution d'un programme pas à pas, points d'arrêts, inspecter les variables pendant l'exécution.
Prévoir les tests unitaires.
Travaux pratiques
Utilisation d'un débogueur pour contrôler l'exécution des programmes.