Smartfuture
Certifications | Training | Technology solutions

Formation & Certification • IT Specialist

Computational Thinking

Cette formation prépare à la certification IT Specialist – Computational Thinking en développant le raisonnement logique, l'analyse de données, la modélisation, la conception d'algorithmes et l'automatisation pour résoudre des problèmes.

Distributeur officiel Certiport | Centre d'examen Certiport | Learn • Practice • Certify

Durée 18,5 h	**Examen** IT Specialist – Com-Thinking
Modalité Distanciel	**Niveau** Fondamental

INSCRIPTION / RÉSERVATION

Je m'inscris maintenant

- **Learn** : Acquisition des concepts clés de la pensée informatique, logique et abstraction selon le référentiel officiel.
- **Practice** : exercices guidés, cas pratiques et simulations d'objectifs d'examen IT Specialist Computational Thinking.
- **Certify** : voucher d'examen IT Specialist Computational Thinking inclus, à passer dans notre centre Certiport.

OBJECTIFS PÉDAGOGIQUES

- Comprendre les types de données et leur représentation.
- Développer le raisonnement logique pour analyser des situations complexes.
- Concevoir et formaliser des solutions sous forme d'algorithmes, puis les automatiser avec séquences, boucles et variables.
- Présenter et améliorer les solutions à l'aide d'artefacts numériques adaptés.

PUBLIC CIBLE

- Pour élèves et étudiants souhaitant renforcer leurs compétences en résolution de problèmes et pensée algorithmique.
- Pour enseignants et formateurs intégrant la pensée informatique dans leurs cours.
- Pour débutants ou personnes en reconversion souhaitant maîtriser les bases de la pensée informatique.

PRÉREQUIS

- Bonne compréhension écrite en français et notions mathématiques de base requises.
- Aucune expérience préalable en programmation n'est nécessaire.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Concepts fondamentaux

1.1 Comprendre et reconnaître les différents types de données

- Distinguer données structurées et non structurées.
- Identifier les principaux types de données : texte, numérique, date/heure, image, audio, etc.
- Comprendre les notions d'encodage des données (binaire, ASCII, mappage de caractères).

1.2 Reconnaître et appliquer le raisonnement logique

- Utiliser les opérateurs booléens et logiques (ET, OU, NON, etc.).
- Appliquer le raisonnement inductif pour dégager des règles à partir d'exemples.
- Identifier les ambiguïtés dans un problème de raisonnement logique.
- Utiliser le raisonnement déductif pour tirer des conclusions à partir de règles établies.

1.3 Expliquer la pensée algorithmique

- Expliquer l'objectif de la pensée algorithmique dans la résolution de problèmes.
- Comprendre le rôle de l'abstraction et de la modélisation dans la simplification de problèmes complexes.
- Expliquer l'intérêt et les possibilités de l'automatisation.

Identifier et collecter les données

2.1 Évaluer les besoins et les données disponibles

- Identifier les données nécessaires pour résoudre un problème donné.
- Évaluer la pertinence de jeux de données existants.
- Déterminer les écarts entre les données disponibles et les besoins réels en données.

2.2 Comprendre la qualité des données

- Comprendre les notions de validité et de fiabilité des données.
- Expliquer ce qu'est le "nettoyage" des données dans un jeu de données (suppression d'anomalies, valeurs manquantes, doublons, etc.).

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Identifier les exigences d'une solution

- Choisir un processus de conception (itératif, incrémental…).
- Identifier les prérequis nécessaires à la mise en œuvre d'une solution.
- Identifier les résultats possibles et les scénarios d'utilisation d'une solution.
- Sélectionner des outils adaptés pour développer une solution (organigrammes, tableurs, pseudo-code, questionnaires, etc.).

Automatiser une solution

Utiliser une séquence d'étapes dans des algorithmes

- Créer une séquence d'instructions claire pour résoudre un problème.
- Évaluer le résultat produit par une séquence d'étapes donnée.
- Reconnaître le moment où il est pertinent de regrouper des étapes dans des procédures ou fonctions réutilisables.

Automatiser les tâches répétitives à l'aide de l'itération

- Reconnaître quand l'utilisation d'une boucle est appropriée.
- Identifier les cas d'utilisation de boucles imbriquées.
- Déterminer le résultat d'un algorithme utilisant une ou plusieurs boucles.
- Créer un algorithme qui utilise l'itération pour automatiser une tâche.

Utiliser des instructions conditionnelles dans les algorithmes

- Reconnaître quand l'utilisation de structures conditionnelles est appropriée.
- Identifier les cas d'utilisation de conditions imbriquées.
- Déterminer le résultat d'un algorithme utilisant des tests conditionnels.
- Créer un algorithme qui utilise des structures de sélection (SI, SINON, etc.).

Utiliser des variables dans les algorithmes

- Reconnaître quand il est nécessaire d'introduire des variables.
- Déterminer le comportement et le résultat d'un algorithme qui manipule des variables.
- Créer un algorithme qui déclare, utilise et met à jour des variables.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Présenter et améliorer une solution

Produire un artefact informatique pour présenter une solution

- Choisir un support de communication adapté au public cible : vidéo, organigramme, maquette HTML, graphique, infographie, etc.
- Créer un artefact numérique original pour communiquer une solution.

Collaborer autour d'artefacts informatiques

- Lire, comprendre et interpréter un design ou un prototype de solution informatique.
- Formuler des retours et des pistes d'amélioration sur un artefact proposé.
- Intégrer des retours collaboratifs afin d'améliorer l'artefact.

Effectuer une conception itérative d'une solution automatisée

- Créer un prototype pour évaluer l'efficacité d'une solution automatisée.
- Comparer l'efficacité de plusieurs solutions possibles à un même problème.
- Diagnostiquer et corriger les erreurs dans une solution automatisée.
- Utiliser des tests itératifs pour améliorer progressivement la solution.