Cycle	Filière	Spécialité
Qualification des cadres d'enseignement	Enseignement secondaire	**INFORMATIQUE**

I. Intitulé du module

ALGORITHMIQUE ET PROGRAMMATION - II

II. Compétence visée

Au terme de ce module, le stagiaire est capable de réinvestir ses acquis relatifs aux concepts avancés en algorithmique et en programmation pour résoudre des problèmes complexes, individuellement ou collectivement, de manière optimale et créative en utilisant des structures de données avancées (fichiers, pointeurs et allocation dynamique de la mémoire, …) et en tenant compte :

- De la démarche algorithmique pour la résolution du problème ;
- Des moyens matériels et logiciels disponibles ;
- Et des usages appropriés des TICE.

III. Objectifs du module

- S'approprier des structures de données avancées adaptées aux spécifications des problèmes posés ;
- Approfondir les connaissances sur les méthodes de conception des algorithmes efficaces et optimaux
- Evaluer les algorithmes
- Apprendre les bases de la programmation en langage python.

IV. Prérequis

- Éléments de base en informatique (système informatique, mémoire, unité arithmétique et logique, périphériques d'entrée, de sortie et de stockage)
- Les concepts de base de base en algorithmique et programmation
- Suites numériques
- Eléments de base en TIC

V. Organisation horaire

Composantes / éléments du module	Volume horaire (VH)					
	Cours	TD	Activités pratiques	Autre (A préciser)	Évaluation	VH global
Volume Horaire	**6**	**12**	**12**	**-**	**4**	**34**
Pourcentage du Volume horaire	**18%**	**35%**	**35%**	**-**	**12%**	**100%**

VI. Contenu du module

Contenu (grandes lignes)	Commentaires
STRUCTURES AVANCEES DE DONNEES ▪ Fichier ▪ Liste ▪ Pile ▪ File	- Présenter les structures concernées à partir des exemples issus de contextes réels.
NOTION DE RECURSIVITE ▪ Définition ▪ Principe de la récursivité ▪ Fonction récursive	- Envisager la correspondance mathématique sur le principe de récurrence. - Présenter quelques exemples de fonctions récursives.
ÉVALUATION DES ALGORITHMES ▪ Notion de complexité algorithmique ▪ Critères de complexité d'un algorithme : temps et espace : - Complexité temporelle : temps nécessaire à l'exécution - Complexité spatiale : espace mémoire nécessaire à l'exécution ▪ Efficacité des algorithmes.	- Adopter la classe inversée et se concentrer lors des séances en présentiel sur le partage, mise en commun et synthèse. - Montrer l'importance d'identifier des algorithmes efficaces. - Comparer et classifier des algorithmes qui résolvent le même problème.
BASES DE LA PROGRAMMATION EN LANGAGE PYTHON ▪ Installation et configuration ▪ Introduction au langage python - Interpréteur et script - Principe de programmation ▪ Bases de programmation - Types de données et variables - Opérateurs et expressions -Séquences (tuple, tableau, chaine de caractères) - Tests et boucles - Fonctions en Python ▪ Gestion des exceptions en Python ▪ Les bibliothèques Python	- Aborder les bases du langage Python en présentant des exemples de codes illustratifs. - Evoquer les différents opérateurs : logiques, booléens, de comparaisons et les opérateurs mathématiques. - Pour optimiser le «Temps machine», les stagiaires sont invités à préparer les programmes hors classe.

VII. Mise en œuvre du module

1) Modalité et activités d'animation

- Présentations
- Démonstrations
- Simulations
- Travaux dirigés

- Travaux pratiques
- Exposés de synthèses, réinvestissement
- Ateliers, projets, recherche documentaire
- Prévoir l'alternance des différentes situations de formation (individuelle, en ateliers, collectives, travaux pratiques)
- Pédagogie par projets aidant les stagiaires à s'auto-documenter, s'auto-former et s'auto-évaluer : Activité de réalisation sur projet

Le formateur adopte des modes de formation diversifiés : Présentiel, à distance, classe inversée, autoformation, …

2) Outils et supports

- Programmes et orientations pédagogiques
- Documents de références
- Ressources numériques
- Outils audiovisuels
- Matériel général d'animation (vidéoprojecteur, flip chart, tableau blanc, …)
- Ordinateurs et simulateurs
- Logiciels et outils adéquats à installer pour la programmation

3) Tâches et activités des bénéficiaires (étudiants, stagiaires, ...)

- Productions individuelles et collectives
- Préparer les programmes hors classe

4) Réinvestissement du module dans la pratique professionnelle (stages)

- Se familiariser avec quelques algorithmes ayant des applications réelles.
- Résoudre des problèmes complexes issus de l'environnement de manière optimales
- Prendre en compte le besoin du marché en termes de langage de programmation et suivre l'évolution technologique
- Exploiter les fonctionnalités de Python pour des fins éducatives.

5) Modalités d'évaluation

- Contrôle continu **(25%) :** Productions des stagiaires, Situations problème visant la mesure du degré d'acquisition des concepts du module, participation, mini-projet …
- Validation du module **(75%) :** Situations complexes à résoudre.