

Intitulé de l'UE	Programmation - Concepts avancés
Section(s)	- (8 ECTS) Bachelier en Informatique orientation Réseaux et Télécommunications / Cycle 1 Bloc 1

Responsable(s)	Heures	Période
Erwin DESMET	84	Quad 2

Activités d'apprentissage	Heures	Enseignant(s)
Algorithmique	24h	Johan DEPRETER
Programmation avancée - théorie	20h	Erwin DESMET
Programmation avancée - travaux pratiques	40h	Erwin DESMET Johan DEPRETER Fabrice SCOPEL

Prérequis	Corequis

Répartition des heures
Algorithmique : 24h de théorie
Programmation avancée - théorie : 20h de théorie
Programmation avancée - travaux pratiques : 40h d'exercices/laboratoires

Langue d'enseignement
Algorithmique : Français, Anglais
Programmation avancée - théorie : Français, Anglais
Programmation avancée - travaux pratiques : Français, Anglais

Connaissances et compétences préalables
[T-PINI-206] Algorithmique Maîtrise de la programmation itérative et impérative
[T-PINI-203] Programmation avancée - théorie Avoir suivi L'UE Programmation - Base et Algorithmique
[T-PINI-204] Programmation avancée - travaux pratiques Avoir suivi L'UE Programmation - Base et Algorithmique

--

Objectifs par rapport au référentiel de compétences ARES

Cette UE contribue au développement des compétences suivantes

- Communiquer et informer
 - Choisir et utiliser les moyens d'informations et de communication adaptés
 - Mener une discussion, argumenter et convaincre de manière constructive
 - Utiliser le vocabulaire adéquat
 - Présenter des prototypes de solution et d'application techniques
 - Utiliser une langue étrangère
- Collaborer à la conception, à l'amélioration et au développement de projets techniques
 - Elaborer une méthodologie de travail
 - Planifier des activités
 - Analyser une situation donnée sous ses aspects techniques et scientifiques
 - Rechercher et utiliser les ressources adéquates
 - Proposer des solutions qui tiennent compte des contraintes
- S'engager dans une démarche de développement professionnel
 - Développer une pensée critique
 - Travailler tant en autonomie qu'en équipe dans le respect de la structure de l'environnement professionnel
- S'inscrire dans une démarche de respect des réglementations
 - Respecter les normes, les procédures et les codes de bonne pratique
- Collaborer à l'analyse et à la mise en œuvre d'un système informatique
 - En choisissant une méthode d'analyse adaptée, exprimer une solution avec les formalismes appropriés
 - Sur base de spécifications issues d'une analyse : (1) développer une solution logicielle ; (2) mettre en œuvre une architecture matérielle

Objectifs de développement durable



Education de qualité

Objectif 4 Assurer l'accès de tous à une éducation de qualité, sur un pied d'égalité, et promouvoir les possibilités d'apprentissage tout au long de la vie

- 4.4 D'ici à 2030, augmenter considérablement le nombre de jeunes et d'adultes disposant des compétences, notamment techniques et professionnelles, nécessaires à l'emploi, à l'obtention d'un travail décent et à l'entrepreneuriat.



Paix, justice et institutions efficaces

Objectif 16 Promouvoir l'avènement de sociétés pacifiques et ouvertes aux fins du développement durable, assurer l'accès de tous à la justice et mettre en place, à tous les niveaux, des institutions efficaces, responsables et ouvertes

- 16.7 Faire en sorte que le dynamisme, l'ouverture, la participation et la représentation à tous les niveaux caractérisent la prise de décisions.

Acquis d'apprentissage spécifiques

[T-PINI-206] Algorithmique

- Enumérer et définir les structures de données usuelles
- Expliquer le fonctionnement des algorithmes vus au cours
- Mettre en œuvre les concepts vu au cours afin de résoudre des problèmes de programmation
- Choisir une structure de données (ou un paradigme de programmation) adaptées afin de résoudre un problème

[T-PINI-203] Programmation avancée - théorie

- Acquérir les bases nécessaires à la conception de programmes
- Appliquer les méthodologies de programmation
- Réalisation de diagrammes UML

[T-PINI-204] Programmation avancée - travaux pratiques

Au terme de ce laboratoire, l'étudiant sera capable de :

- Schématiser un diagramme UML modélisant un objet en utilisant des concepts de Programmation Orientée Objet (tels que accesseurs/mutateurs, visibilité, héritage,...);
- Etablir un code informatique à partir de ce diagramme UML vers un langage cible ;
- Etablir des classes et leurs composants
- Créer une application graphique à répondre à un cahier des charges spécifié (structuration, emploi de contrôles, gestion de l'apparence, ergonomie, gestion des sollicitations, interaction avec fichiers,...)

Contenu de l'AA Algorithmique

- Notions des structures de données usuelles : liste (doublement) chaînée/circulaire, ensemble, pile, file, dictionnaire, arbre, tas, graphe
- Algorithmes de manipulation des structures de données usuelles (ajout, suppression, modification)
- Algorithmes de tri : insertion, bulle, rapide, fusion, par tas
- Notions de récursivité (terminale, mutuelle, ...)
- Notions de la théorie des graphes (parcours, recherche de chemin, coloration)

Contenu de l'AA Programmation avancée - théorie

- Programmation événementielle
- Programmation objet
- UML

Contenu de l'AA Programmation avancée - travaux pratiques

Apprentissage du langage Python dans le paradigme POO :

- Présentation et familiarisation à la modélisation UML
- Manipulation des contrôles principaux (python)
- Présentation et familiarisation au concept de "programmation événementielle"
- Présentation et familiarisation au concept de "programmation Orientée Objet"
- Réalisation de petits projets avec interfaces graphiques qui permettent de se confronter aux concepts cités ci-dessus. (Exemples de réalisations: calculatrice, éditeur de texte, gestion d'une clientèle basée sur un fichier, calendrier, jeu de société, ...)

Méthodes d'enseignement

Algorithmique : cours magistral, travaux de groupes, approche interactive

Programmation avancée - théorie : cours magistral, approche par projets, approche interactive, approche par situation problème

Programmation avancée - travaux pratiques : travaux de groupes, approche par projets, approche avec TIC, utilisation de logiciels

Supports

Algorithmique : copies des présentations, syllabus, notes de cours

Programmation avancée - théorie : syllabus

Programmation avancée - travaux pratiques : notes d'exercices, protocoles de laboratoires

Ressources bibliographiques de l'AA Programmation avancée - théorie

- Desmet Erwin, Notes de cours «Programmation avancées », HEH - Campus Technique, 2023.

Ressources bibliographiques de l'AA Programmation avancée - travaux pratiques

- Chapelle Joakim, Depreter Johan et Desmet Erwin , Notes de cours «Programmation avancées », HEH - DST, 2023.

--

Évaluations et pondérations	
Évaluation	Note globale à l'UE
Langue(s) d'évaluation	Français, Anglais
Méthode d'évaluation	<p>En ce qui concerne la notation totale de cette UE, nous utiliserons la règle suivante :</p> <ul style="list-style-type: none">• Partie labo réussie mais pas le reste : 3/20• Partie théorique réussie mais pas le labo : 3/20 <p>Réussite de l'EI (réussites des 3 AA) -> note pondérée (/20) selon la répartition ci-dessous :</p> <ul style="list-style-type: none">- Epreuve d'examen pour les 3 AA : Algorithmique 25 % - POO 20 % - Laboratoire 45%- Evaluation continue 10 % non rejouable.
Report de note d'une année à l'autre pour l'AA réussie en cas d'échec à l'UE	
Algorithmique : non Programmation avancée - théorie : non Programmation avancée - travaux pratiques : non	

Année académique : **2023 - 2024**