HEH be Sciences et technologies

8a avenue Maistriau
7000 Mons

www.heh.be

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

Responsable(s)	Heures	Période
Johan DEPRETER	100	**Quad 2**

Activités d'apprentissage	Heures	Enseignant(s)
Algorithmique	25h	**Jean-Sébastien LERAT**
Hackathon	20h	**Julien DE BODT** **Antoine MALAISE** Johan DEPRETER Erwin DESMET Fabrice SCOPEL
Programmation avancée - théorie	20h	**Antoine MALAISE**
Programmation avancée - travaux pratiques	35h	**Julien DE BODT** Johan DEPRETER Erwin DESMET Fabrice SCOPEL

Prérequis	Corequis

Répartition des heures
Algorithmique : 15h de théorie, 10h de travaux
Hackathon : 20h de séminaires
Programmation avancée - théorie : 20h de théorie
Programmation avancée - travaux pratiques : 35h d'exercices/laboratoires

Langue d'enseignement
Algorithmique : Français, Anglais
Hackathon : Français
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

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 oeuvre 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

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

Au terme de 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, visiblilités, héritage,...);
- Etablir un code informatique à partir de ce diagramme UML vers un langage cible ;
- Créer une application graphique à répondant 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, supression, modification)
- Algorithmes de tri : insertion, bulle, rapide, fusion, par tas
- Notions de la théorie des graphes (parcours, recherche de chemin, coloration)
- Notions de récursivité (terminale, mutuelle, ...)

Contenu de l'AA Hackathon

Réalisation d'un travail collaboratif visant à résoudre un problème informatique donné dans des conditions proches de celles d'un projet d'entreprise. Le groupe d'étudiant devra

- élaborer son cahier des charges (canevas imposé);
- décomposer les tâches à accomplir;
- planifier son emploi du temps;
- construire son application répondant à la demande;
- défendre le travail réalisé ainsi que les choix opérés.

Le travail réalisé devra être remis et fera ensuite objet d'une évaluation par les enseignants.

La participation individuelle effective à l'activité interviendra également dans l'évaluation.

Contenu de l'AA Programmation avancée - théorie

- Etude des contrôles utilisateurs (Feuille, boutons, zone de saisie, menu, case à cocher,...)
- Programmation événementielle
- Programmation objet
- Lecture et écriture dans des fichiers

Contenu de l'AA Programmation avancée - travaux pratiques

Apprentissage du langage C# et utilisation de l'outil de développement d'applications Microsoft Visual Studio :

- Présentation et familiarisation à le modélisation UML
- Manipulation des contrôles principaux (C# / .Net)
- Présentation et familiarisation au concept de "programmation événementielle"
- Présentation et familiarisation au concept de "programmation Orientée Objet"
- Réalisation de petit 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

Hackathon : approche par projets, approche interactive, approche par situation problème, approche avec TIC, étude de cas, utilisation de logiciels

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

Hackathon : copies des présentations

Programmation avancée - théorie : syllabus

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

Ressources bibliographiques de l'AA Programmation avancée - théorie

- Malaise Antoine, Notes de cours «Technique informatique », HEH - Campus Technique, 2014.
- Deitel H. M & Deitel P.J., « C# How to program », Prentice-Hall,2004

Ressources bibliographiques de l'AA Programmation avancée - travaux pratiques

- Malaise Antoine, Notes de cours «Technique informatique », HEH - Campus Technique, 2014.
- Deitel H. M & Deitel P.J., « C# How to program », Prentice-Hall,2004
- Jérôme Hugon**,** "C# 5 *Développez des applications Windows avec Visual Studio 2012*", Editions ENI, 2012 (ISBN : 9782746077164)
- Ressources en ligne : ” http://dotnet.developpez.com/csharp/ ”

Évaluations et pondérations	
Évaluation	Note globale à l'UE
Langue(s) d'évaluation	Français, Anglais
Méthode d'évaluation	Chaque acquis d’apprentissage sera évalué de manière autonome et aura une note comprise entre 0 et 20. En ce qui concerne la notation totale de cette UE, nous utiliserons la régle suivante : 0 acquis validé -> 0/20 1 acquis validé -> 3/20 2 acquis validés -> 5/20 au moins 3 acquis validés -> note pondérée (/20) selon la répartition ci-dessous : - Prog. avancée - Théorie 20% - Prog. avancée - Travaux pratiques 40% - Algorithmique 20% - Hackaton 20% (non remédiable)

Report de note d'une année à l'autre pour l'AA réussie en cas d'échec à l'UE

Algorithmique : **non**
Hackathon : **non**
Programmation avancée - théorie : **non**
Programmation avancée - travaux pratiques : **non**

Année académique : **2020 - 2021**