Descriptif de module 63-11

Domaine : Economie & Services
Filière : Informatique de gestion

1. Intitulé de module — Fondement de la programmation — 2026-2027

Code :
63-11

Type de formation :
☒ Bachelor ☐ Master ☐ MAS ☐ DAS ☐ CAS ☐ Autres :

Niveau :
☒ Module de base
☐ Module d'approfondissement
☐ Module avancé
☐ Module spécialisé
☐ Autres :

Caractéristique :
☒ Module dont l'échec peut entrainer l'exclusion définitive de la filière selon l'art.15, al.1 des directives cadres "statut des étudiants-e-s"

Type :
☒ Module principal
☐ Module lié au module principal
☐ Module facultatif ou complémentaire
☐ Autres :

Organisation temporelle :
☒ Module sur 1 semestre
☐ Module sur 2 semestres
☒ Semestre d'automne
☐ Semestre de printemps
☐ Autres :

2. Organisation

Crédits ECTS
5

Langue principale d'enseignement :
☒ **Français**
☐ Allemand
☐ Autres :
☐ Italien
☐ Anglais

3. Prérequis

☐ Avoir validé le module
☐ Avoir suivi le module
☒ Pas de prérequis
☐ Autres :

4. Compétences visées / Objectifs généraux d'apprentissage

L'étudiante ou l'étudiant doit être capable, en fin de module, de justifier des compétences professionnelles suivantes :

- Maîtriser les fondements algorithmiques
- Savoir résoudre un problème méthodologiquement, décomposer sa solution en algorithme, décrire l'algorithme en pseudo-code et le formuler dans un langage de programmation

5. Objectifs détaillés des enseignements

L'étudiante ou l'étudiant doit être capable, en fin de module, de justifier des compétences professionnelles suivantes :

- Maîtriser les notions fondamentales de l'algorithmique et de la logique booléenne pour les mettre en œuvre avec un langage opérationnel (Java)
- Exprimer formellement la solution d'un problème au moyen des structures de données et des schémas algorithmiques élémentaires
- Mettre en œuvre les formants algorithmiques usuels pour élaborer la solution d'un problème selon une approche itérative

6. Plan et chapitres des cours

Au travers de ce cours, les étudiant-es apprendront les fondements de la programmation.
Ainsi, le cours est composé de cinq thèmes :

1) **Variables et opérateurs**
   a. Types simples
   b. Types complexes
   c. Opérateurs logiques

2) **Instructions de base**
   a. Tests de logique booléenne
   b. Les boucles (Pour, Tant Que, Boucles imbriquées, …)

3) **Les sous-programmes**
   a. Fonctions et modules
   b. Passages de paramètres

4) **Les structures de données**
   a. Tableaux uni- et multidimensionnels
   b. List et ArrayList
   c. Dictionnaires (HashMap)

5) **Traitements de fichiers**
   a. Chargement / entrées
   b. Déchargement / sorties
   c. Traitement d'erreurs

Dans le cadre du cours qu'est-ce qui est traité au niveau durabilité ?
L'usage de l'Intelligence Artificielle est gourmand en énergie, mais dans le cadre de l'apprentissage de la programmation, vous pouvez utiliser l'IA pour lui demander d'analyser vos algorithmes ou votre pseudo-code pour identifier comment les rendre plus efficients. Une boucle mal conçue ou un algorithme non optimisé consomme inutilement de la puissance de calcul (CPU) et de la mémoire (RAM). L'IA peut être un outil pertinent pour vous aider à adopter très tôt les bons réflexes de l'éco-conception logicielle.

7. Forme du cours et méthodes pédagogiques

Organisation

- Durée du module (contrôles continus inclus, examens exclus) : 15 semaines
- Quatre heures de cours théoriques avec des séquences pratiques d'algorithmique et programmation par semaine
- Une heure de cours théorique de logique booléenne
- Une heure de pratique en laboratoire « Algorithmique, Programmation et Logique »
- En dehors de ces séances, l'assistant est également à disposition des étudiant-e-s sur rendez-vous pour répondre à des questions personnelles concernant le cours et/ou les exercices pratiques proposé.

Forme

- Cours théoriques et séquences pratiques (laboratoires) impliquant une participation des étudiant-e-s.

Travail personnel

- En plus des heures de cours, nous attendons de chaque étudiant-e qu'il-elle consacre un certain temps chaque semaine à la consolidation et à l'étude de ses notes de cours ainsi qu'à la résolution des problèmes proposés dans les travaux pratiques
- L'étudiant-e est encouragé-e à prendre en charge son propre processus d'apprentissage.

Utilisation de l'IA dans le cadre du module
☒ Dans le cadre du cours ☒ Dans le cadre des laboratoires ☐ Dans le cadre des évaluations

Description globale : L'objectif de ce module est de développer votre propre esprit logique et vos compétences en programmation. Bien que les outils d'Intelligence Artificielle (ChatGPT, GitHub Copilot, Claude, etc.) soient puissants, ils peuvent freiner votre apprentissage s'ils effectuent le travail à votre place.

En règle générale, toute utilisation de l'IA, lorsqu'elle est autorisée, **doit être documentée de façon transparente** dans les documents remis ainsi que dans le développement de la solution (par exemple, via des commentaires dans le code source).

Dans le cadre du cours

L'utilisation de l'IA doit être la plus limitée possible. Elle ne doit pas remplacer votre participation ou votre réflexion personnelle.

Utilisations autorisées :

- **Compréhension conceptuelle :** Demander à l'IA d'expliquer différemment ou de vulgariser un concept théorique vu en classe que vous n'avez pas compris.
- **Génération d'exercices :** Demander à l'IA de vous générer des exercices d'entraînement supplémentaires (et leurs corrigés) pour réviser la matière à la maison.

Dans le cadre des laboratoires

L'objectif des laboratoires est de vous confronter aux problèmes et d'apprendre de vos erreurs. L'utilisation de l'IA doit y être **exceptionnelle et minimale**.

- **Ce qui est interdit :** Utiliser l'IA pour générer tout ou partie du pseudo-code, code et des logigrammes de vos laboratoires, ou pour résoudre l'architecture ou l'algorithmique du programme à votre place.
- **Ce qui est toléré :** Utiliser l'IA de manière ponctuelle pour vous débloquer sur l'explication d'un message d'erreur spécifique ou pour comprendre le fonctionnement d'une fonction précise.
- **Rappel :** Toute aide de ce type doit être explicitement mentionnée en commentaire dans votre code.

Dans le cadre des évaluations

- L'utilisation de toute forme d'Intelligence Artificielle générative est **strictement interdite** sous toutes ses formes (ex : examens écrits, examens pratiques sur machine, projets notés).
- Tout recours à ces outils lors d'une évaluation sera considéré comme une tentative de fraude ou de plagiat et sanctionné comme tel.

8. Modalités d'évaluation et de validation

Présence :
La présence à toutes les activités d'enseignement est obligatoire et constitue une condition de validation du module. Tout absence doit être justifiée au près du secrétariat dans les 48 heures suivant l'absence.
Les étudiant-e-s dont le taux de présence est inférieur à 75% ne sont pas autorisé-e-s à réaliser l'examen final ce qui entraîne la note de 1 à l'examen pouvant entraîner l'échec au module.

Acquis : A-E
Remédiation : Fx
Répétition : F

L'évaluation du module se fera en principe de la manière suivante :

Contrôle continu : 50%	**Examen : 50%**
1 contrôle continu théorico-pratique individuel de 90' en semaine 5 — 25% 1 contrôle continu pratique de 90' en semaine 12 — 25%	Examen pratique de 180'

Remédiation / Fx :
Un module pour lequel l'étudiant-e obtiendrait une note entre 3.5 et 3.9 peut faire l'objet d'une remédiation, conformément à l'art. 29 al. 1 du Règlement sur la Formation de base en HES-SO.
Si la remédiation est validée la note de 4 sera attribuée.
Si la remédiation n'est pas validée le module sera répété.

Répétition / F :
Un module pour lequel l'étudiant-e obtiendrait une note inférieure ou égale à 3.4 doit être répété.

L'étudiant-e qui n'obtient pas les crédits attribués à un module doit le répéter dès que possible, conformément à l'art. 30 al. 1 du Règlement sur la formation de base en HES-SO.