

Descriptif de module 63-11

Domaine : Economie & Services
Filière : Informatique de gestion

1. Intitulé de module **Fondement de la programmation** **2022-2023**

Code :
63-11

Niveau :

- Module de base
 Module d'approfondissement
 Module avancé
 Module spécialisé
 Autres :

Type :

- Module principal
 Module lié au module principal
 Module facultatif ou complémentaire
 Autres :

Type de formation :

- Bachelor Master MAS DAS CAS Autres :

Caractéristique :

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

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 Italien
 Allemand Anglais
 Autres :

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, décomposer sa solution en algorithme et la 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 (ex. Python)
- 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. Passage de paramètres
- 4) **Les structures de données**
 - a. Tableaux uni- et multidimensionnels
 - b. Tableaux de chaînes de caractères
 - c. Tableaux associatifs
- 5) **Traitements de fichiers**
 - a. Chargement / entrées
 - b. Déchargement / sorties

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 proposés.

Forme

- Cours théoriques et séquences pratiques 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.

8. Modalités d'évaluation et de validation

Acquis : A-E
Remédiation : Fx
Répétition : F

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

Contrôle continu : 50% 1 contrôle continu pratique individuel de 90' — 30% 1 contrôle continu pratique en binôme —20%	Examen : 50% Examen pratique de 180'
--	--

La réalisation personnelle, attestée par l'assistant, de 75% des travaux pratiques (avec doc) proposés est exigée pour avoir le droit de se présenter à l'examen.