

Année universitaire 2024/2025

Talents Mathématiques-Informatique - 2e année bis de Licence

Crédits ECTS : 42

LES OBJECTIFS DE LA FORMATION

Le parcours Talents permet aux étudiants sportifs de haut niveau ou artistes (musique, danse, arts, art dramatique) de suivre exactement le programme de L1-L2 tout en bénéficiant d'un aménagement : étalement des cours sur six semestres au lieu de quatre. Ils poursuivent en Licence de Mathématiques ou en Licence d'Informatique.

Connaissances de base en mathématiques,
Connaissance de base en informatique,
Connaissance de base en économie.

POURSUITE D'ÉTUDES

Cette formation peut être naturellement prolongée par la Licence Mathématiques Appliquées ou Informatique des Organisations puis par un Master dans le département MIDO.

PROGRAMME DE LA FORMATION

- Semestre 5T
 - UE Obligatoires Semestre 5T Maths-Info
 - Algèbre linéaire 3
 - Anglais 3
 - Architecture des ordinateurs
 - Programmation C
 - Algorithmique et programmation 3
- Semestre 6T
 - UE Obligatoires Semestre 6T Maths-Info
 - Algèbre 4 et méthodes numériques
 - Anglais 4
 - Programmation système
 - Fonctionnal programming

DESCRIPTION DE CHAQUE ENSEIGNEMENT

Algorithmique et programmation 3

ECTS : 5

Description du contenu de l'enseignement :

Chacun des points suivants sera présenté et expérimenté en langage Python :

1. Algorithmes et fonctions logarithmes : logarithmes naturels dans les appels récursifs où dans les boucles type série harmonique, preuves courtes des propriétés de base des logarithmes. Notations asymptotiques et arrondis récursifs.
2. Complexité : algorithmes en $T(n)=aT(n-b) + \text{poly}(n)$, et application aux implémentations exponentielle/linéaire de Fibonacci et à l'algorithme d'Euler-Bachet-Bezout.
3. Récursivité de la forme $T(n)=aT(n/b) + \text{poly}(n)$: (rappel tri fusion), preuve courte du "master theorem", calcul rapide de complexité à partir du cas n puissance de b .
4. Performance des algorithmes : application du "master theorem" à la conception d'algorithmes de multiplication rapide d'entiers (Karatsuba), et de matrices (Strassen).
5. Tri : variétés du concept de complexité (pire cas, moyenne, expression des données) avec les algorithmes classiques de tri (rappel: insertion, dénombrement, tas)
6. Force brute : algorithmes énumératifs, application à la résolution de systèmes d'équations et aux placements de reines sur échiquiers $n \times n$.

Compétence à acquérir :

Fondements mathématiques de la complexité algorithmique et idée précises, avec connaissance profondes des exemples emblématiques, de ses paradigmes centraux. Maîtrise des mécanismes de base du langage Python.

Algèbre 4 et méthodes numériques

ECTS : 4

Description du contenu de l'enseignement :

1. Résolution numérique de systèmes linéaires (méthodes directes et itératives).
2. Calcul numérique de valeurs propres (méthode de la puissance).
3. Résolution numérique d'équations scalaires non linéaires (méthodes d'encadrement et de point fixe, méthode de la sécante).
4. Interpolation polynomiale.
5. Formules de quadrature interpolatoires.

Compétence à acquérir :

Présentation de méthodes numériques de résolution et d'éléments d'analyse numérique. Mise en œuvre : utilisation de Python Numpy et Jupyter (travaux pratiques et projet).

Algèbre linéaire 3

ECTS : 8

Description du contenu de l'enseignement :

1. Réduction des endomorphismes : diagonalisation et trigonalisation.
2. Formes bilinéaires.
3. Formes quadratiques
4. Espaces euclidiens : produit scalaire, norme euclidienne, orthogonalité, bases orthonormées et procédé de Gram-Schmidt, projection orthogonale, isométries vectorielles et endomorphismes auto-adjoints.

Compétence à acquérir :

Réduction des endomorphismes, formes bilinéaires et quadratiques, espaces euclidiens.

Anglais 3

ECTS : 2

Description du contenu de l'enseignement :

Contenu : professionnels, culturels, d'actualité et de société

Forme : débats, jeux de rôles, quiz et activités ludiques

Méthodologie : prise de parole en public, travail sur l'expression orale

Thématiques au programme: Communication & customer service, Team building & building relationships, Money & finance

Compétence à acquérir :

Savoir s'exprimer à l'oral

Améliorer ses compétences langagières et communicationnelles

Enrichir son vocabulaire

Développer sa créativité

Travailler en équipe

Mode de contrôle des connaissances :

100% contrôle continu

-3 notes : test écrit +présentation orale + note d'oral

(test écrit de 2e chance en fin de semestre ouvert à tous les étudiants qui le souhaitent)

-travail individuel hebdomadaire (grammaire, vocabulaire, compréhension et expression écrites)

Anglais 4

ECTS : 2

Description du contenu de l'enseignement :

Contenu : professionnels, culturels, d'actualité et de société

Forme : débats, jeux de rôles, quiz et activités ludiques

Méthodologie : prise de parole en public, travail sur l'expression orale

Thématiques au programme: Sports, Job satisfaction, success & failure, Crisis management

Compétence à acquérir :

Savoir s'exprimer à l'oral

Améliorer ses compétences langagières et communicationnelles

Enrichir son vocabulaire

Développer sa créativité

Travailler en équipe

Mode de contrôle des connaissances :

100% contrôle continu

-3 notes : test écrit +présentation orale + note d'oral

(test écrit de 2e chance en fin de semestre ouvert à tous les étudiants qui le souhaitent)

-travail individuel hebdomadaire (grammaire, vocabulaire, compréhension et expression écrites)

Architecture des ordinateurs

ECTS : 4

Description du contenu de l'enseignement :

Histoire de l'informatique. Représentation des nombres et arithmétique.

Circuits logiques. Structure générale d'un ordinateur

L'unité centrale : instructions, registres, pipeline, interruptions L'assembleur.

Les mémoires : hiérarchie, mémoire électronique, mémoire cache, mémoire de masse Les entrées/sorties.

Performances d'un ordinateur.

Compétence à acquérir :

Comprendre la structure interne d'un ordinateur à travers l'étude de ses différents composants : microprocesseur, mémoire, entrées/sorties ; acquérir les notions de base en langage machine : instruction, adressage, assembleur.

Functionnal programming

ECTS : 4

Description du contenu de l'enseignement :

La notion de fonction comme valeur, et les "high-order functions".

La récursivité dans le contexte de la programmation fonctionnelle et les questions de complexité associées.

Les systèmes de types qui permettent la manipulation de fonctions et de fonctions partielles (la notion de Currying)

La récursivité dans le contexte des types. Les types infinis, les structures de données associées et leur manipulation.

La théorie mathématique de la programmation fonctionnelle (lambda-calcul, théorème de Church-Rosier) et la théorie mathématique de la vérification du comportement d'un programme (sémantiques).

Toutes ces sujets seront élaborés avec des exercices de programmation en Haskell.

Compétence à acquérir :

L'objectif de ce cours est de familiariser les étudiants avec les principes de la programmation fonctionnelle en utilisant le langage Haskell. La programmation fonctionnelle est un paradigme moderne de programmation qui permet la conception à la fois rapide et fiable des applications complexes. Les notions de la programmation fonctionnelle, comme les expressions lambda, sont actuellement prévalents dans la plupart des langages de programmation modernes, comme Java, C++, JavaScript, etc. Le but de ce cours est d'aider les étudiants à les maîtriser en utilisant un langage purement fonctionnel (Haskell). De plus, le cours couvrira le système de types de Haskell, montrera comment un tel système de types peut nous aider à concevoir des programmes dont on peut vérifier le bon fonctionnement de manière formelle, et donnera les bases mathématiques de la théorie de la programmation fonctionnelle.

Programmation C

ECTS : 4

Description du contenu de l'enseignement :

Types et expression.

Structures de contrôle.

Fonctions.

Tableaux et pointeurs.

Structures.

Préprocesseur.

Entrées/sorties.

Compétence à acquérir :

Apprentissage du langage C de base et évolué.

Programmation système

ECTS : 4

Description du contenu de l'enseignement :

Rôle du système d'exploitation et de son interface de programmation.

Étude et mise en pratique de l'utilisation d'un système Unix.

Étude et mise en pratique de la programmation Shell.

Étude des principaux appels systèmes de l'interface Posix (gestion de fichiers, processus).

Réalisation d'exercices simples mettant en œuvre chacun de ces appels système.

Réalisation d'un exercice complet combinant tous ces appels système.

Compétence à acquérir :

Ce cours est orienté vers l'utilisation du système d'exploitation par le développeur. Il s'agit donc d'étudier l'interface de programmation d'un système d'exploitation, l'interface Posix des systèmes Unix en l'occurrence. On vise ainsi à donner un sens concret à la notion de système et à son utilisation par les développeurs. Le cours comporte une partie pratique importante d'utilisation du systèmes Linux et de sa programmation.

