MI-Informatique S2

Infos pratiques

> **ECTS** : 6.0
> **Nombre d'heures** : 55.0
> **Langue(s) d'enseignement** : Français
> **Niveau d'étude** : BAC +1
> **Période de l'année** : Enseignement second semestre
> **Méthodes d'enseignement** : En présence
> **Forme d'enseignement** : Cours magistral et Travaux dirigés
> **Ouvert aux étudiants en échange** : Oui
> **Campus** : Campus de Nanterre
> **Composante** : Sciences économiques, gestion, mathématiques et informatique
> **Code ELP** : 5E2MIINF

Présentation

Ce cours prolonge les bases de la programmation en abordant la conception de programmes structurés, robustes et efficaces. Il porte notamment sur la décomposition d'un problème en sous-problèmes, l'utilisation des fonctions, la récursivité, les pointeurs, les structures, l'allocation dynamique, les fichiers et les principales structures de données. Les notions sont principalement illustrées par un langage de programmation (C, python ou autre), avec une attention particulière portée aux choix de conception, à l'organisation du code et à la gestion de la mémoire.

Le cours vise aussi à montrer comment les outils actuels fondés sur les LLM et les agents peuvent assister le travail d'un étudiant en algorithmique et en programmation. Ces outils seront utilisés pour rechercher de l'information, comprendre un message d'erreur, expliquer un fragment de code, proposer une première solution, produire des tests ou accompagner la réalisation d'un projet. Leur usage sera cependant présenté comme une aide au raisonnement et au développement, et non comme un substitut à la compréhension personnelle.

Une place importante sera accordée à la rigueur de programmation, à la vérification du code produit, à la capacité à tester ses fonctions et à l'analyse critique des réponses fournies par les outils d'IA.

Objectifs

À l'issue du cours, les étudiants devront être capables de :

- Concevoir un algorithme pour résoudre un problème non trivial ;
- Décomposer un problème complexe en fonctions et sous-problèmes plus simples ;
- Utiliser la récursivité et les principes du diviser pour régner ;
- Manipuler correctement les pointeurs, les structures et l'allocation dynamique ;
- Lire et écrire des données dans des fichiers ;
- Choisir une structure de données adaptée à un problème donné ;
- Produire un programme clair, modulaire, testé et maintenable ;
- Utiliser un LLM ou un agent pour assister la recherche d'information, la compréhension du code, le débogage et la réalisation d'un projet ;
- Vérifier de manière critique les propositions générées par un outil d'IA.

Évaluation

Modalités : Mixte : CC + CT
SESSION 1 :
Contrôle Continu
• Type : Écrit, Projet, TP
• Durée : --
• Précisions : Le contrôle continu représente 50 % de la note finale
et comprend un devoir écrit ou sur machine comptant pour 25 %, un projet comptant pour 15 %, ainsi qu'une évaluation du présentiel, de l'implication et des activités réalisées en TD ou en TP comptant pour 10 %.
Contrôle Terminal
• Type : Écrit
• Durée :1h30

• Précisions : L'examen final représente 50 % de la note finale.
Régime Dérogatoire
• Type : Écrit
• Durée : 1h30
SESSION 2 :
• Type : Écrit
• Durée : 1h30

 # Les différentes épreuves écrites, qu'il s'agisse de l'examen final, du devoir, de l'examen terminal en régime dérogatoire ou de la seconde session, ont une durée indicative de 1h30, susceptible d'être ajustée en fonction des contraintes pédagogiques, matérielles ou logistiques, dans le respect des M3C applicables. Les absences, les aménagements éventuels et les autres situations particulières sont traités conformément aux M3C en vigueur.

Utilisation de l'intelligence artificielle :
Pour cet EC, l'usage de l'intelligence artificielle (IA) générative pour aider à la réalisation des travaux soumis à évaluation ***est interdite***, *sauf éventuelles* exceptions dûment spécifiées par l'enseignant responsable du cours. Dans le cadre de ces éventuelles exceptions, les principes suivants doivent être respectés.
Les résultats fournis par l'IA doivent constituer un matériau pour votre réflexion, et toujours faire l'objet d'une réappropriation et d'une reprise critique.
Tous les usages doivent être documentés dans une section dédiée à la fin de votre travail, à l'exception des usages de recherche web augmentée, de correction orthographique et syntaxique. Cette section doit permettre à votre lecteur d'évaluer la manière dont vous avez travaillé avec l'IA et mobilisé cette ressource au service d'un travail personnel.
L'intégration directe de contenus engendrés par l'IA doit être faite sous le régime de la citation.
Toute utilisation non déclarée ou non autorisée d'un outil d'IA pourra être considérée comme un manquement aux règles d'intégrité académique.
Il est également demandé aux étudiants de ne pas transmettre à des outils d'IA des données confidentielles ou personnelles concernant l'université, les enseignants, les entreprises partenaires ou d'autres étudiants.

Compétences visées

- Concevoir et expliquer des algorithmes.
- Programmer des solutions structurées.
- Utiliser les fonctions pour organiser et tester le code.
- Comprendre et manipuler la mémoire, les adresses et les pointeurs.
- Définir et utiliser des types structurés.
- Gérer dynamiquement la mémoire lorsque la taille des données n'est pas connue à l'avance.
- Manipuler des fichiers pour conserver ou exploiter des données persistantes.
- Mettre en œuvre des structures de données classiques.
- Évaluer la correction, la lisibilité et l'efficacité d'un programme.
- Mobiliser des LLM et des agents comme outils d'assistance au développement, tout en gardant la maîtrise de la conception, de la validation et de la correction du code.

Contact(s)

> Nicolas Rauwel
Responsable pédagogique
nrauwel@parisnanterre.fr