

Licence Mathématiques

Algorithmique et programmation 1

Responsable	Descriptions	Informations
	Code : SMI3U12	Composante : Faculté des Sciences
	Nature : Unité d'enseignement	
	Domaines : Sciences et Technologies	

LANGUE(S) D'ENSEIGNEMENT

Français

CONTENU

Cette UE se base sur l'UE de Programmation en Python qui aura été suivie en portail Descartes dont l'objectif est d'utiliser le paradigme de programmation orienté objet, à l'aide du langage Python.

Programme

- Remédiation de méthodologie de la programmation orientée objet en Python (surtout pour les nouveaux entrants)
- Programmation (orientée objets) en Python
 - Gain d'autonomie en programmation orientée objets : écrire soi-même des classes en Python et les utiliser de façon fluide pour structurer du code
 - Utilisation d'exceptions
 - Contrôle de boucles : continue, break
 - Héritage de classes
 - Surcharge d'opérateurs simples
 - Utilisation des bibliothèques numpy et matplotlib
- Algorithmique
 - Preuve de correction d'algorithmes
 - Preuve de complexité d'algorithmes
- Algorithmes et programmes permettant d'illustrer les concepts théoriques :
 - Méthode des trapèzes pour un calcul approché d'intégrales sur un segment (utiliser numpy pour les subdivisions puis visualiser le schéma via matplotlib)
 - Écriture d'une classe Polynome avec visualisation en matplotlib et recherche dichotomique pour le zéro d'une fonction polynomiale (évoquer les problèmes de convergence éventuels dus à des approximations flottantes)
 - Algorithme de multiplication matricielle avec sa complexité
 - Algorithmes d'élevation à la puissance (dont l'exponentiation rapide et sa complexité)
 - Intersection de deux courbes (à nouveau, évoquer les problèmes d'approximation flottante)

étudiants provenant d'autres L1 n'en auront vraisemblablement pas vu. Il est donc important de prévoir des feuilles de TD/TP adaptées au niveau de chacun pour assurer une progression de toutes et tous. Les exemples d'algorithmes et programmes cités plus haut sont une liste non exhaustive d'idées permettant d'illustrer les concepts de programmation et d'algorithmique à appréhender.

Compétences

À la fin du cours, on s'attend à ce que les étudiants validant le cours soient en mesure de :

- programmer un algorithme en Python avec structuration en classes
- convaincre de la correction d'un algorithme simple
- estimer la complexité d'un algorithme simple
- faire preuve d'esprit critique sur les erreurs d'approximation flottante inhérente au calcul numérique
- comparer deux algorithmes réalisant la même tâche pour choisir le meilleur selon les cas
- utiliser les bibliothèques numpy et matplotlib pour réaliser des calculs et visualiser des courbes

MODALITÉS D'ORGANISATION

12h de cours, 6h de TD, 12h de TP

VOLUME HORAIRE

- Volume total: 30 heures
- Cours magistraux: 12 heures
- Travaux dirigés: 6 heures
- Travaux pratiques: 12 heures

CODES APOGÉE

- SMI3U12A [ELP]
- SMI3U12L [ELP]
- SMI3U12C [ELP]
- SMI3U12T [ELP]

M3C

Aucune donnée M3C trouvée

POUR PLUS D'INFORMATIONS

[Aller sur le site de l'offre de formation...](#)



COMPÉTENCES À ACQUÉRIR

Connaissances du cours

Ce cours est l'occasion de réaliser de la différenciation pédagogique puisqu'il faudra composer avec des étudiants ayant suivi le cours de portail Descartes de programmation orientée objets (en Java pour les étudiants de L2 en 2024-2025, puis en Python ensuite), alors que les