

Intitulé de l'Unité d'Enseignement : Programmation Orientée Objet en Python.

Nombre de crédits de l'UE	3
Semestre où l'enseignement est proposé	M1 - S1

a) Objectifs pédagogiques

Cet enseignement présentant les concepts fondamentaux de la Programmation Orientée Objet (Classes, Objets, Héritage). Les objectifs de cette UE sont :

- d'acquérir la maîtrise des concepts fondamentaux de la POO (Classes, Objets, Héritage) et des mécanismes associés (encapsulation, abstraction) qui permettront aux étudiants d'écrire des programmes structurés
- d'apprendre à modéliser sous la forme de classes un cahier des charges.

Le cours se structure autour d'exemples de modélisation en UML ainsi que du code Python (commenté) correspondant afin de permettre aux étudiants d'appréhender les notions essentielles. Des Travaux Dirigés et Pratiques leur permettent ensuite de se familiariser à leur mise en œuvre.

b) Compétences acquises par l'étudiant

A l'issue de cet enseignement, les étudiants seront capables de modéliser un cahier des charges simple en un diagramme UML et de proposer l'implémentation correspondante en Python.

c) Contenu de l'enseignement

Plan du Cours :

Cours 1: Rappel d'algorithmie et de syntaxe Python

- conditionnelles, boucles,
- fonctions, bibliothèques de base (maths)
- tableaux (et listes)

Cours 2: Classes et Objets

- attributs, méthodes, encapsulation,
- constructeur, destructeur
- association / agrégation,
- surcharge d'opérateurs (égalité, affichage, ...)
- attributs et méthodes de classes

Cours 3: Héritage et Abstraction

Les TDs et TPs s'articuleront autour de la problématique de la recherche du zéro d'une fonction par dichotomie ou par l'algorithme de Newton. Les étudiants commenceront par une implémentation (non objet) dans le cas des polynômes puis intégreront progressivement le code réalisé dans une modélisation objet de plus en plus complexe permettant de traiter diverses fonctions (sinus, exponentielle, somme de fonctions, ...).

Dès le début de l'UE, un sujet de projet est dévoilé aux étudiants qui devront le modéliser et proposer une implémentation pour la fin du semestre. Ce projet est réalisé en groupe et évalué.

Un travail personnel sera demandé aux étudiants en début d'UE pour acquérir une bonne connaissance de la syntaxe Python (non objet).

d) Prérequis

Notion d'algorithmie

e) Modalités de contrôle des Connaissances

2 notes d'écrits (2 QCM, 20% + 1 examen final, 40%), 1 note de projet (40%).

Organisation pédagogique

Enseignements présentiels	Volume horaire total	Effectif par groupe
Cours	8	Total
Travaux Dirigés	6	
Travaux Pratiques	12	26

Projet		
Autre		

Cadencement temporel de l'UE :

N°	Cours	TD	TP	CC
S1	C1			
S2	C2		TP1 (2h)	
S3	C3	TD1		
S4		TD2	TP2 (2h)	
S5	C4		TP3 (2h)	QCM1
S6		TD3		
S7			TP4 (4h)	
S8				QCM2
S9				
S10				
S11				ER1

N°	Cours	TD	TP	CC
S1	Rappel			
S2	Classes & Objets		Prise en main	
S3	Classes & Objets	Algo + UML		
S4		Classes & Objets	Fonctions & Tableaux	QCM1 Python
S5	Héritage		Classes & Objets	
S6		Héritage		
S7			Héritage	
S8				QCM2 POO
S11				ER2