

# Conception et programmation orientées Objet

Niveau d'étude  
**Bac +4**

ECTS  
**6 crédits**

Composante  
**Sciences Fondamentales  
et Appliquées**

Période de l'année  
**Semestre 1**

## En bref

- # **Langue(s) d'enseignement:** Français, Anglais
- # **Méthode d'enseignement:** En présence
- # **Organisation de l'enseignement:** Formation initiale
- # **Forme d'enseignement :** Total
- # **Ouvert aux étudiants en échange:** Non

## Présentation

### Description

Cette UE a pour but de faire découvrir les bases de la conception orientée-objet avec la notation UML, et de la programmation orientée-objet en langage Java. Elle est enseignée en anglais, et utilise les principes du cours inversé, de l'apprentissage et de l'évaluation entre pairs, et de l'apprentissage par problèmes.

### Objectifs

*Connaissances visées*

- \* Principes de l'orienté-objet

- \* Principe du développement piloté par les tests (test-driven development)

### Heures d'enseignement

Conception et programmation orientées Objet - CM	CM	2h
Conception et programmation orientées Objet - TD	TD	8h
Conception et programmation orientées Objet - TP	TP	20h
Conception et programmation orientées Objet - Plateforme	Plate forme en autonomie	20h

### Pré-requis nécessaires

Bases de la programmation impérative et de l'algorithmique

### Programme détaillé

- \* Découverte des concepts de l'orienté-objet
- \* Mise en œuvre des principes de base
- \* Approfondissement des principes de base de l'orienté-objet
- \* Initiation à la programmation des interfaces graphiques

- \* Mise en œuvre intégrative de la conception d'une application orientée-objet en développement piloté par les tests

## Compétences visées

### *Compétences :*

- \* Conception orientée-objet (COO)
  - \* Expliquer les concepts de l'orienté-objet
  - \* Abstraire une classe
  - \* Expliquer les relations entre classes, dont l'héritage
  - \* Concevoir un modèle de classes en UML
  - \* Décrire la dynamique d'une application avec UML
  - \* Tester une application
  - \* Construire une application orientée-objet
- \* Programmation orientée-objet (POO) en Java
  - \* Programmer une classe en Java
  - \* Programmer les relations entre classes en Java
  - \* Programmer l'héritage en Java
  - \* Mobiliser les bases de l'algorithmique objet en Java
  - \* Programmer des interfaces graphiques simples avec Java
  - \* Tester avec JUnit
  - \* Mettre au point un programme en Java

---

## Infos pratiques