

## Programmation orientée objet avancée

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Informatique et systèmes de communication
<b>Orientations</b>	Informatique logicielle (ISCL)
<b>Mode de formation</b>	Plein temps

### Informations générales

Nom	: Programmation orientée objet avancée
Identifiant	: POA
Années académiques	: 2021-2022, 2022-2023
Responsable	: Pier Donini
Charge de travail	: 90 heures d'études
Périodes encadrées	: 48 (= 36 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours						24			
Laboratoire						24			

### Connaissances préalables recommandées

L'étudiant-e doit maîtriser les concepts de base de la programmation orientée objet.

L'unité POO permet d'acquérir ces connaissances.

### Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- utiliser les concepts OO du langage C++ et énoncer le vocabulaire associé aux spécificités de ce langage ;
- expliquer et utiliser les spécificités du langage C++ : constructeurs, constructeur de copie, destructeur, allocation dynamique, surcharge des opérateurs, héritage simple et multiple, polymorphisme, méthode et classe virtuelles, , RTTI (runtime type information) ;
- expliquer et utiliser les concepts avancés du C++ : généricité, itérateurs, liste d'initialiseurs, boucle foreach, foncteurs, pointeurs intelligents ;
- expliquer et utiliser les concepts avancés de Java : interfaces fonctionnelles, expressions lambda, streams ;
- comparer les spécificités des langages Java et C++.

A l'issue des travaux pratiques en laboratoire, l'étudiant-e sera en outre capable de :

- maîtriser la syntaxe et les concepts avancés de POO des langages étudiés (Java et C++) ;
- concevoir et réaliser un diagramme de classes pour modéliser un cas concret et en réaliser l'implémentation.

## Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

**Cours:** 24 périodes

- Etude des particularités du langage C++ : référence, pointeur et variable objet, constructeurs, constructeur de copie, destructeur, allocation dynamique, surcharge des opérateurs, static, polymorphisme et héritage simple et multiple, méthode et classe virtuelles, RTTI 8
- Etude des concepts avancés du C++ : généricité, itérateurs, liste d'initialiseurs, for\_each, foncteurs, pointeurs intelligents 8
- Etude des concepts avancés de Java : interfaces fonctionnelles, expressions lambda, streams 8

**Laboratoire:** 24 périodes

- Mise en oeuvre en C++ de différents constructeurs et destructeurs, avec utilisation d'allocation de mémoire dynamique et surcharge d'opérateurs 8
- Conception UML et implémentation en C++ d'applications gérant des collections d'objets et mettant en oeuvre les concepts de généricité, polymorphisme et de liaison dynamique 10
- Mise en oeuvre des expressions lambda lambda et des streams Java sur des cas concrets 6

## Bibliographie

Aucune information

## Contrôle de connaissances

**Cours :** l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 1.5 période.

**Laboratoire :** ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

**Examen :** L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 60 minutes.

Matériel autorisé :

- Information communiquée directement par l'enseignant.

## Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5