

## Programmation orientée objet 2

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Informatique
<b>Orientation</b>	Logiciel (IL)
<b>Mode de formation</b>	Plein temps

### Informations générales

Nom:	:	Programmation orientée objet 2
Identifiant:	:	POO2
Années académiques	:	2016-2017, 2017-2018, 2018-2019, 2019-2020, 2020-2021
Responsable:	:	Pier Donini
Charge de travail:	:	90 heures d'études
Périodes encadrées:	:	64 (= 48 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours						32			
Laboratoire						32			

### Connaissances préalables recommandées

L'étudiant-e doit maîtriser les concepts de base de la programmation orientée objet. L'unité POO1 permet d'acquérir ces connaissances.

### Objectifs

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

- utiliser les concepts POO 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, espace de nommage, approche de l'encapsulation, surcharge des opérateurs, héritage simple et multiple, polymorphisme, méthode et classe virtuelles, fonction, méthode et classe amies ;
- expliquer et utiliser les concepts avancés du C++ 11 : généricité, itérateurs, liste d'initialiseurs, boucle foreach, foncteurs, pointeurs intelligents ;
- expliquer et utiliser les concepts avancés de Java 8 : interfaces fonctionnelles, lambda expressions, 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 8 et C++11) ;
- 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:** 32 périodes

- Etude des particularités du langage C++ : référence, pointeur et variable objet, constructeurs, constructeur de copie, destructeur, surcharge des opérateurs, static, polymorphisme et héritage simple et multiple, méthode et classe virtuelles, RTTI, fonction, méthode et classe amies 16
- Etudes des concepts avancés du C++ 11 : généricité, itérateurs, liste d'initialiseurs, for\_each, foncteurs, pointeurs intelligents 8
- Etude des concepts avancés de Java 8 : interfaces fonctionnelles, lambda expressions, streams 8

**Laboratoire:** 32 périodes

- Prise en main de la syntaxe et des concepts de base orientés objet du langage C++ 4
- Mise en oeuvre en C++ de différents constructeurs et destructeurs, avec utilisation d'allocation de mémoire dynamique (strings ou tableaux de taille dynamique) et surcharge d'opérateurs 10
- Conception UML et implémentation en C++ d'applications gérant des collections d'objets au moyen de la STL et mettant en oeuvre les concepts de généricité, polymorphisme et de liaison dynamique 12
- Mise en oeuvre des lambda expressions et des streams de Java 8 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 2 périodes.

**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

Fiche validée le 16.08.2016 par Donini Pier