

CS-112(j)

Programmation orientée objet

Doeraene Sébastien

Cursus	Sem.	Type
Mathématiques	BA2	Obl.

Langue d'enseignement	français
Coefficient	4
Session	Eté
Semestre	Printemps
Examen	Pendant le semestre
Charge	120h
Semaines	14
Heures	3 hebdo
Cours	1 hebdo
Projet	2 hebdo
Nombre de places	

Résumé

Ce cours approfondit les connaissances en programmation présentées dans le cours ICC du 1er semestre. L'accent est mis sur la conception et la spécification de programmes (en Python) via l'approche « orientée objet » et la réalisation d'une mini-application dans un projet réalisé par binôme.

Contenu

- Fondements de l'approche orientée objet pour la modularité : structure de classe, encapsulation, méthodes, héritage, polymorphisme.
- Méthodes de développement de programmes : documentation, tests, développement collaboratif
- Bibliothèques usuelles d'outils (bibliothèques standard et tierces)
- Programmation générique
- Les concepts théoriques introduits lors des cours magistraux seront mis en pratique dans le cadre d'exercices sur machines et par le biais de la réalisation d'un projet.

Mots-clés

langage Python, programmation orientée-objet, conception de logiciel

Compétences requises**Cours prérequis obligatoires**

CS-119(l) Information, Calcul, Communication

Concepts importants à maîtriser

bases de programmation procédurale, par exemple en C++, Java ou Python

Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Modéliser, à l'aide de la couche orientée objet du langage Python, une situation modérément complexe du monde réel décrite en français
- Analyser un problème complexe pour le décomposer en sous-problèmes
- Concevoir des algorithmes résolvant des tâches modérément complexes
- Transcrire un algorithme en son programme équivalent en Python

- Tester l'adéquation du résultat d'un programme par rapport à la tâche demandée
- Evaluer l'impact des algorithmes choisis et de leur implémentation sur les performances
- Réaliser de façon autonome et collaborative une application de taille moyenne
- Analyser du code Python avancé pour en décrire le résultat ou le corriger s'il est erroné

Compétences transversales

- Persévérer dans la difficulté ou après un échec initial pour trouver une meilleure solution.
- Accéder aux sources d'informations appropriées et les évaluer.
- Planifier des actions et les mener à bien de façon à faire un usage optimal du temps et des ressources à disposition.
- Gérer ses priorités.
- Utiliser une méthodologie de travail appropriée, organiser un/son travail.

Méthode d'enseignement

Séances ex cathedra complétant un cours en ligne (MOOC), travaux pratiques sur ordinateur

Travail attendu

Participation au cours, résolutions d'exercices, réalisation d'un projet par groupe de deux, travail personnel « à la maison ».

Méthode d'évaluation

Contrôle continu :

- une série notée intermédiaire pour 17% ;
- un examen écrit (2 h), pour 33% ;
- un projet par groupe de 2, pour 50%.

Encadrement

Office hours	Non
Assistants	Oui
Forum électronique	Oui

Ressources

Service de cours virtuels (VDI)

Oui

Bibliographie

J.-C. Chappelier, C++ par la pratique, PPUR 2012 (3e édition)

Ressources en bibliothèque

- [J.-C. Chappelier, C++ par la pratique](#)

Sites web

- <https://www.coursera.org/learn/programmation-orientee-objet-cpp/>
- <http://www.progmaph.ch/>

Liens Moodle

- https://go.epfl.ch/CS-112_g

Vidéos

- <https://www.coursera.org/learn/programmation-orientee-objet-cpp/>
- <https://go.epfl.ch/poo-videos>