

Introduction à la Programmation : Présentation du cours

Laboratoire d'Intelligence Artificielle
Faculté I&C

Objectifs du cours « Introduction à la Programmation »

Objectif : apprendre à programmer (en Java)

- ▶ connaître les **bases de la programmation**
- ▶ se familiariser avec les **concepts orientés objet** au travers du langage Java
- ▶ **mettre en pratique des concepts algorithmiques** présentés formellement dans le cours AICC : récursion, structures de données abstraites, algorithmes de tri

Présenter le cours :

- ▶ Objectifs (**Quoi ?**)
- ▶ Administration (**Comment ?**)

Présentation générale du cours (1)

Public : Cours obligatoire pour les étudiants de 1^{er} semestre.
Pas de connaissances préalables requises en informatique.

Langue : Français

Moyens : Concepts théoriques introduits via deux **MOOCs** : vidéos et quizzes
Compléments sous la forme de **cours magistraux** ex-cathedra (**mercredi 14¹⁵–15⁰⁰**)
Mise en pratique, de manière guidée, lors de **séances d'exercices** sur ordinateurs (**vendredi 13¹⁵–16⁰⁰**).

Suite : Cours « *Pratique de la programmation orientée-objet* » (CS108, M. Schinz)

Site Moodle du cours :

<https://moodle.epfl.ch/course/view.php?id=14847>

- Contenu :
- ▶ pour une vue d'ensemble voir la ressource «Calendrier et support»
 - ▶ chaque semaine a un contenu dédié

Enseignants : Jamila Sam et Rafael Pires

Encadrement : voir la section «Contacts» de Moodle.

Conseil : Lisez avec attention la documentation à votre disposition

Niveau du cours

- ▶ Vous êtes débutant ?
 - ▶ cours exigeant mais ne nécessitant pas de connaissances préalables
- ▶ Vous avez déjà programmé un peu ?
 - ▶ cours parfois facile, mais apprenez à « bien » programmer
- ▶ Vous êtes déjà très avancé ?
 - ▶ Si vous pensez ne pas devoir venir au cours, alors soit, mais...
Attention ! : l'examen portera sur le contenu du cours !



Seul point d'entrée à utiliser pour toutes les ressources du cours !

Beaucoup d'informations et de ressources y sont fournies.

- ☞ Familiarisez vous rapidement avec sa structure
- ☞ Apprenez à y naviguer efficacement
- ☞ Consultez régulièrement les « Annonces du cours »

Quelques éléments clés

1. **Couplage à deux MOOCs** durant 10 semaines du cours
2. **Mini-projets** : résolution de problèmes
3. cette année le cours commence à être co-enseigné
 - ☞ Rafael Pires

Couplage aux MOOCs (1)

MOOC d'initiation à la programmation en Java :

www.coursera.org/learn/initiation-programmation-java/

MOOC d'introduction à la programmation orientée-objets en Java :

www.coursera.org/learn/programmation-orientee-objet-java/

Notre cours dispose de ses propres transparents et séries d'exercices

☞ Sur-ensemble du MOOC

☞ **Inscrivez-vous !**

Certificats (payants) : **en aucun cas obligatoires pour ce cours**

Couplage aux MOOCs (3)

- ▶ Avant le cours : visionner les vidéos, faire les quizzes et parcourir les transparents
- ▶ Cours ex-cathedra : résumé et approfondissements
 - ☞ en général, 1 heure seulement
- ▶ Exercices : mise en pratique (comportant le devoir MOOC)



Attention ! : notre cours a un **rythme plus rapide** que celui du MOOC

☞ Certaines semaines couvrent deux semaines du MOOC à elles seules

Couplage aux MOOCs (2)

Matériel MOOC utilisé :

1. Vidéos
2. Quizzes
3. Vidéos
4. Devoirs (mais ne comptent pas)

☞ à considérer comme des exercices supplémentaires et comme aide à la préparation aux tests

Couplage aux MOOCs (4)

Charge de travail :

- ▶ 1 à 2 heures de cours ex-cathedra : **récapitulation et approfondissements** ;
- ▶ 3 heures d'exercices en salle de TP : **mise en pratique** ;
- ▶ 5 à 6 heures **de travail à la maison** :
 - ▶ 1 :30-1 :45 sur les vidéos de la semaine suivante
 - ▶ 0 :15-0 :30 sur les quizzes de la semaine suivante
 - ▶ 3 à 4 heures pour commencer à préparer la série d'exercices de la semaine en cours, finaliser celle de la semaine passée et faire les devoirs du MOOC.

L'apprentissage via le MOOC convient généralement bien si vous débutez.

Si vous trouvez ce format trop lent et que vous avez déjà des notions vous pouvez utiliser les **BOOCs**.

Il s'agit de documents multi-média publié par les PPUR qui :

- ▶ synthétisent le contenu des MOOCs,
- ▶ et permettent d'accéder par simple clic à des moments choisis des vidéos.

À utiliser aussi comme support de révision pour les examens après utilisation des MOOCs

Mini-projets

Apprentissage de la programmation via la **résolution de problèmes concrets**

- ☞ Le mini-projet 1 est un élément **d'auto-évaluation**, la note automatiquement générée ne compte pas dans la moyenne
- ☞ **Le mini-projet 2 est noté**
 - ▶ Mini-projet 1 : « Éléments de cryptographie et de stéganographie »
 - ▶ Mini-projet 2 : « Jeux sur grille »
- ▶ Mon collègue [Rafael Pires](#) est en charge du mini-projet 1.
- ▶ **Ne négligez pas le reste des apprentissage au profit des mini-projets.**

Java (LTS 21)

- ☞ Les tutoriels d'installation sont dans la première série

IntelliJ

- ☞ Vous pouvez utiliser d'autres outils si vous le souhaitez, comme Eclipse par exemple, mais **moins de support** de la part des assistants.

Le cours du second semestre utilisera Java 21 et IntelliJ



Attention ! Les correcteurs du MOOC utilisent des versions plus anciennes de Java (ils sont en cours de mise à jour)

Mini-projet 1 : cryptographie et stéganographie (1)

But du mini-projet 1 :

- ▶ montrer que l'on peut déjà faire des choses intéressantes et utiles, même avec un bagage basique en programmation ;
- ▶ disposer d'un outil pour auto-évaluer son niveau (correcteur automatique) ;
- ▶ commencer à apprendre à travailler en binôme ([git](#) comme outil de collaboration).

Projets des années précédentes (qui étaient notés) :

- ▶ traitement d'images ;
- ▶ reconnaissance de lettre manuscrites ;
- ▶ génération de QR-code ;
- ▶ reconnaissance d'empreintes digitales.
- ▶ ...

Mini-projet 1 : cryptographie et stéganographie(2)

But du projet : écrire un programme capable de chiffrer des données selon différents algorithmes et de cacher ces messages chiffrés dans des images.

- ▶ Les techniques de chiffrements (cryptographie) sont omniprésentes en informatique pour sécuriser l'échange de données.
 - ▶ La stéganographie permet de cacher un message dans une image.
- ☞ Le but du mini-projet est de vous faire faire une petite incursion dans ces univers

Projection des cours

Afin de vous permettre un maximum de flexibilité, les cours seront **projetés en visio-conférence** via le service Mediaspace (voir l'adresse sur Moodle).

Les vidéos mises à disposition **peuvent avoir été enregistrées les années précédentes** si elles sont jugées bien refléter le cours donné.

Il est vivement recommandé de suivre le cours et le TP aux horaires prévus même si vous êtes hors campus pour le cours !

- ▶ Vous assurez ainsi la régularité de vos acquis et de votre progression (clé essentielle de réussite au cours).
- ▶ Vous pouvez bénéficier d'un maximum de support de la part des assistants pendant les TPs.

Mini-projet 2 : encore un jeu

But du mini-projet 2 : expérimenter les atouts de l'approche orientée objet

- ▶ en produisant soi-même du code
- ▶ mais aussi en intégrant du code existant (maquette de base fournie)

Les jeux sont un terrain d'expérimentation très fertile avec beaucoup d'ouvertures pour personnaliser le résultat du projet !

- ☞ Sensiblement plus exigeant que le mini-projet 1
- ☞ **Ne négligez pas le tutoriel qui précède le début du codage**

Forum de discussion (Ed)

Ce forum accessible via la page Moodle du cours :

- ▶ Constitue le **moyen d'interaction central en dehors des heures de contact**
- ▶ Permet de poster des questions qui seront prises en charge de façon régulière par les assistants :
 - ▶ Les questions peuvent aussi être posées anonymement :-)
 - ▶ Elles peuvent être adressées en particulier aux instructeurs ou à un instructeur en particulier
- ▶ vous pouvez l'utiliser dès qu'une question se pose à vous **pendant toute la semaine**.
 - ☞ Nous ferons tout notre possible pour vous répondre dans les meilleurs délais (maximum un jour d'attente)

Salles de travaux pratiques

Dans les salles de TP, vous disposez de :

- ▶ **64 postes fixes et 64 postes flottants** (INF3)
- ▶ deux fois **128 postes flottants** (INF1 et INF2)
 - ↳ Vous pouvez y connecter vos propres laptops
- ▶ trois salles sans machines sont aussi réservées en appoint : **INF019**, **INF213**, et **INM203** (pourront être utilisées selon la demande).
- ▶ les salles CO20-23 offrent des postes de travaux fixes qui sont aussi utilisables (utilisez alors les VMs linux IC-CO-IN-SC-MA-2024-Fall ou IC-CO-IN-SC-INJ-2024-Fall)
- ▶ vous pouvez utiliser ces salles ainsi que les salles INF1, INF2 et INF3 en tout temps pour peu qu'elles ne soient pas prises par une autre TP

©EPFL 2024
J. Sam

EPFL

CS-107 – Cours 1 : Informations générales – – 21 / 33

Séances Pratiques

- ▶ **Réservation des salles INF1, INF2 et INF3** ainsi que de salles d'appoint (**INF019, INF213 et INM203**)
Ces salles sont réservées pour vous les **vendredis 13¹⁵–16⁰⁰**.
- ▶ **Réservation des salles CO20 et CO21-22 en appoint**
Ces salles sont réservées pour vous les **vendredis 13¹⁵–14⁰⁰** pour les 3 premières semaines du semestre.
- ▶ **Travail en dehors des séances de contact**

Nécessaire pour finaliser les séries et se préparer au prochain cours !

La présence aux exercices est **vivement recommandée** si vous débutez.

La présence d'assistants en CO et salles d'appoint est à définir en fonction des besoins.

©EPFL 2024
J. Sam

EPFL

CS-107 – Cours 1 : Informations générales – – 22 / 33

Séances Pratiques (2)

- ▶ Exercices catégorisés de 0 à 3 dans un ordre croissant de difficulté
 - ↳ **niveau des connaissances en programmation très hétérogène en première année**
- ▶ Niveau 0 = reprises commentées d'exemples vus en cours
- ▶ Niveau 1 = « mise en jambe »
- ▶ Niveau 2 et 3 = approfondissements et consolidation
- ▶ Devoirs MOOC = entraînement aux tests

©EPFL 2024
J. Sam

EPFL

CS-107 – Cours 1 : Informations générales – – 23 / 33

Séances Pratiques (3)



Attention ! : les séries sont parfois longues

- ↳ But : vous fournir du matériel d'entraînement et de préparation

Il ne vous est pas demandé de tout faire pendant les heures d'exercices !



Attention ! : C'est seulement en faisant **un maximum d'exercices** que vous serez bien préparés pour l'examen et le projet du second semestre.

- ↳ Comme pour les autres cours, il faut travailler en dehors des heures d'exercices !
- ↳ Voir le document sur la pédagogie du cours, disponible depuis la rubrique "Références" du site web, pour plus de détails

©EPFL 2024
J. Sam

EPFL

CS-107 – Cours 1 : Informations générales – – 24 / 33

Un support hebdomadaire est offert aux étudiants du cours CS107 en **salle INF3**, avec 2 à 3 assistants à disposition.

Ces séances sont principalement destinée à apporter du support **aux plus débutants** d'entre vous ou ponctuellement en cas d'incompréhension d'un concept donné ou pour résoudre un problème technique d'installation

🗨️ **vendredi 18⁰⁰–19³⁰**

Premiers TP



Attention ! : il est impératif de procéder à l'installation de Java et d'un outil de développement intégré (IntelliJ) sur vos propres machines.

🗨️ **But** : mise en place de votre environnement de travail pour le cours de programmation

Les tutoriaux IntelliJ/Eclipse peuvent être faits sur n'importe quel type de machine contrairement à l'exercice sur l'environnement UNIX/Linux (qui constitue une ouverture de base à ce genre d'OS).

Inscrivez-vous au MOOC si vous voulez accéder aux vidéos, quizzes et devoirs.

- ▶ Contenu délibérément allégé
- ▶ De nombreux exercices du premier TP sont spécifiques aux machines de l'EPFL
- ▶ Le but principal est de vous aider à avoir une installation vous permettant de réaliser les exercices dans de bonnes conditions pendant le semestre.

Support de cours

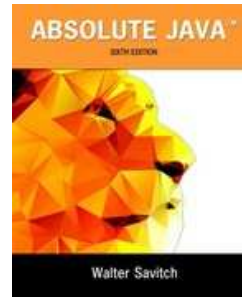
- ▶ **Transparents** mis à disposition via le site Moodle avec parfois des références complémentaires ([vidéos et quizzes des MOOCs](#), bibliographie et/ou hyperliens Internet)
- ▶ **Énoncés des exercices** disponibles sur le site Web en fin de semaine
- ▶ **Corrigés des exercices** disponibles sur le site Web en fin de semaine

Ces éléments constituent une **base suffisante** pour ce cours !

L'ouvrage de référence suivant peut être un complément utile :

Walter Savitch

Absolute Java, 6th Edition



Plusieurs autres ouvrages peuvent aussi convenir (conseils d'achats sur la page web du cours)

Notes et examens : Mini-projets

- ▶ A programmer pendant environ 2 semaines pour le premier et 3 pour le second
- ▶ Le mini-projet1 est un élément d'auto-évaluation dont la note automatiquement attribuée ne compte pas.
- ▶ le mini-projet2 est noté.
- ▶ Faits par groupes de deux (inscription sur Moodle le moment venu).

Date butoir du mini-projet 1 lundi 04 novembre 2024
Date du rendu du mini-projet 2 jeudi 19 décembre 2024

Chacun des modules de programmation de cette année (« *Introduction à la Programmation* » et « *Pratique de la programmation orientée-objet* ») vous vaudra une note séparée.

La note de ce cours est calculée comme suit :

- ▶ Mini-projet 2 (environ 3 semaines) ⇒ coef. 60%
- ▶ Test écrit (2 heures) ⇒ coef. 40%

Notes et examens : Examen final

- ▶ Le semestre sera clôturé par un examen écrit, individuel, portant sur le contenu du cours et les séances d'exercices.
- ▶ **La présence à l'examen est obligatoire.**

Date : **Samedi 14 Décembre 2024**

Auditoires : communiqués en temps voulu

Comment nous contacter ?

Plusieurs moyens pour contacter les enseignants, assistants et étudiants-assistants pour poser des questions sur le cours ou les exercices :

- ▶ Durant les séances d'exercices :
 - ▶ c'est le moyen le plus direct si vous êtes autorisé(e) à venir
- ▶ Par l'intermédiaire du forum Ed du cours (accessible via Moodle ou directement depuis l'adresse du forum)
- ▶ en adressant un email

✉ pour des aspects administratifs uniquement : Jamila Sam

Les contacts par téléphone, emails ou visites devront être **strictement réservés aux cas urgents !**