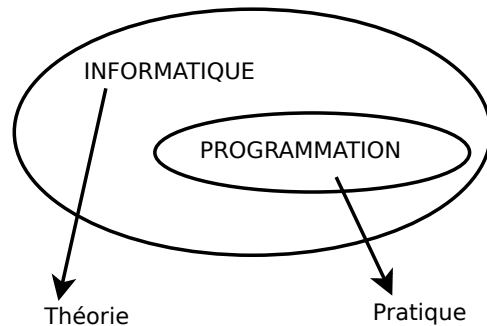


Information Calcul et Communication : Présentation générale

Jamila Sam

Laboratoire d'Intelligence Artificielle
Faculté I&C

Structure générale du cours ICC



Partie théorique → Rafael Pires
Partie pratique → Jamila Sam

- ▶ Cours obligatoire pour les étudiants du 1er semestre de la section des Sciences de la Vie.
- ▶ Connaissances supposées acquises : aucune

Plan

- ▶ Présenter le cours :
 - ▶ Objectifs (« **Quoi ?** »)
 - ▶ Administration (« **Comment ?** »)
- ▶ Introduire la programmation : notion d'algorithme
- ▶ Présenter l'environnement de travail pour la partie pratique

Objectifs de la partie théorique

- ▶ Présenter l'informatique en tant que discipline scientifique
- ▶ Exposer ses fondements conceptuels
- ▶ Développer la pensée algorithmique
 - 📖 « Computational thinking »
- ▶ Expliquer les bases de fonctionnement du monde numérique
- ▶ Sensibiliser à la sécurité dans ce monde numérique
 - 📖 D'avantage lors du cours de vendredi !

Objectifs de la partie pratique

La programmation n'est qu'une partie de l'informatique mais une partie importante et une discipline en tant que telle.

Buts de la partie pratique :

1. Apprendre à **programmer**
savoir les bases et connaître correctement au moins un langage
 - ☞ pratique sur le langage C++
2. Appréhender par la pratique certaines des problématiques conceptuelles vues dans la partie théorique
3. Savoir comment **utiliser** un **ordinateur** (sous Linux) dans le cadre du développement de programmes

Organisation de la partie pratique

Langue : Français

Moyens :

Concepts introduits ou complémenté lors de **cours** magistraux ex-cathedra

☞ **Lundi 12¹⁵–13⁰⁰**

mis en pratique, de manière guidée, lors de **séances d'exercices** sur machines

☞ **Mardi 08¹⁵–10⁰⁰**

Compléments en lignes : **vidéos** et **quizzes** (disponibles pour 8 semaines du cours).

Organisation de la partie théorique

Langue : Français

Moyens :

Concepts introduits lors de **cours** magistraux ex-cathedra

☞ **Vendredi 9¹⁵–11⁰⁰**

mis en pratique, de manière guidée, lors de **séances d'exercices** sur papier

☞ **Vendredi 15¹⁵–16⁰⁰**

Séances d'appui

Un support additionnel sera offert certaines semaines aux étudiants du cours ICC en **salle C020**.

Toute question pourra y être posée (théorie ou pratique)

☞ **Lundi 17³⁰–19⁰⁰**

☞ **Jeudi 18⁰⁰–19⁰⁰**

Ces séances seront annoncées en temps voulu.

Il faudra vous y inscrire au préalable sur le site Moodle afin que nous puissions gérer convenablement les besoins en assistants !

Accès au matériel du cours

- Logistique (salles, horaires, planning, forums etc.)
- Contenu (transparents, exercices etc.)

📁 Site Moodle du cours :

<https://moodle.epfl.ch/course/view.php?id=15751>

Forum Ed

<https://edstem.org/eu/courses/2252/discussion/>

- 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)

Projection des cours

Afin de vous permettre un maximum de flexibilité, les cours seront **projetés en visio-conférence** (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 pour le présentiel, 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.

Interaction avec les enseignants

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 (accessible via Moodle, ou directement depuis l'adresse du forum)
 - 📁 moyen idéal pour partager/discuter, faire part de ses difficultés et les résoudre
- N'hésitez pas à en faire usage !**
- par email s : 📁 pour des aspects administratifs uniquement : Jamilia Sam
 - 📁 mais pour les cas généraux, préférez le forum.

Les contacts personnels par email, téléphone ou visites devront être **strictement réservés aux cas personnels et/ou urgents !**

Support de cours

Tout le matériel est accessible via le **site Moodle** du cours :

- ▶ **Transparents**
parfois enrichis de notes techniques (**mini-références**)
détaillant certains concepts évoqués pendant le cours, en
particulier les éléments du langage C++
également parfois des références complémentaires
(bibliographiques et/ou hyperliens Internet)
- ▶ **Énoncé des exercices**
disponibles sur le site Moodle en fin de semaine.
- ▶ **Corrigé des exercices**
disponibles sur le site Moodle en fin de semaine suivante.
- ▶ Pour la partie pratique : **Vidéos et quizzes**
offerts par le cours massif en ligne (**MOOC**) associé au cours
(voir plus loin)

©EPFL 2025-26
Jamila Sam
& Jean-Cédric Chappelier



ICC (partie programmation) – Cours 1 : Présentation du cours – 13 / 35

Livres ? (2)

Le matériel du MOOC, les transparents et divers supports du site Moodle devraient constituer une **documentation suffisante** pour la partie programmation !

Si vous souhaitez la compléter, l'ouvrage suivant est également recommandé pour la partie pratique :

J.-C. Chappelier & F. Seydoux
C++ par la pratique – recueil d'exercices corrigés et aide-mémoire, EPFL Press, 4^{ème} édition.
Empruntable en version électronique auprès de la bibliothèque ou en vente pour environ 35 CHF.



©EPFL 2025-26
Jamila Sam
& Jean-Cédric Chappelier



ICC (partie programmation) – Cours 1 : Présentation du cours – 15 / 35

Livre ? (1)

Pour la partie théorique, le livre suivant est recommandé :



André Schiper (éditeur), EPFL Press, 2^{ème} édition 2018
(prix étudiant 35 CHF)

©EPFL 2025-26
Jamila Sam
& Jean-Cédric Chappelier



ICC (partie programmation) – Cours 1 : Présentation du cours – 14 / 35

Couplage au MOOC (1)

MOOC d'initiation à la programmation en C++ :

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

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

👉 Sur-ensemble du MOOC

Matériel MOOC utilisé :

1. Vidéos
2. Quizzes
3. Devoirs (mais ne comptent pas)
 - 👉 à utiliser pour se préparer aux tests

©EPFL 2025-26
Jamila Sam
& Jean-Cédric Chappelier



ICC (partie programmation) – Cours 1 : Présentation du cours – 16 / 35

Couplage au MOOC (2)

- ▶ Avant le cours : visionner les vidéos, faire les quizzes et comprendre certains exercices de niveau 0
- ▶ Cours ex-cathedra : résumé et approfondissements
📺 **seulement une heure**
- ▶ Exercices : mise en pratique
- ▶ Certificats (payants) : **en aucun cas obligatoires pour ce cours**

Les BOOCs

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** (référencés sur le site Moodle du cours).

Il s'agit de documents multi-média publié par les EPFL Press 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

Couplage au MOOC (3)

Charge de travail pour la partie pratique :

- ▶ 45 mn de cours ex-cathedra : **récapitulation et approfondissements** ;
- ▶ 1h45 d'exercices en salle de TP : **mise en pratique** ;
- ▶ environ 3h30 heures **de travail à la maison** :
 - ▶ 1h30-1h45 sur les vidéos de la semaine suivante
 - ▶ 0 :15-0 :30 sur les quizzes de la semaine suivante
 - ▶ environ 1 :30 heures pour commencer à préparer la série d'exercices de la semaine en cours, finaliser celle de la semaine passée.

Notes et examens

Les épreuves de contrôle continu seront les suivantes :

- ▶ Mid-term I (théorie + quiz «sécurité») **individuel**
- ▶ Mini-projet (théorie + pratique) **en binôme, ≈ 3 semaines**
- ▶ Mid-term II (théorie + pratique) **individuel**

Calcul de la note

- La note finale du semestre, N , est calculée comme suit :

$$N = (35\% * N_{Mid_term_1} + 10\% * N_{mini_projet} + 55\% * N_{Mid_term_2})$$

- Les notes intermédiaires ne sont pas arrondies.
- Les cours ICC et «Programmation orientée-objet» sont indépendants. La moyenne arrondie de chaque cours est transmise au SAC à la fin de chaque semestre.

Notes et examens Mid-term II

Le semestre sera clôturé par un examen **écrit** portant sur le contenu du cours et les séances d'exercices, aussi bien pour la partie théorique que pratique.

Réalisé individuellement

Documentation partielle autorisée (fiches résumé)

Date :

Samedi 13 Décembre

Notes et examens Mid-term I

Votre progression à mi-semestre sera évaluée par un examen **écrit** portant sur le contenu du cours et les séances d'exercices, pour la partie théorique.

Réalisé individuellement

Documentation partielle autorisée (fiches résumé)

Date :

Vendredi 31 Octobre (ou Samedi 1)

Notes et examens Mini-projet

La dernière semaine du projet sera consacrée au rendu du mini-projet.

Date :

Mercredi 17 Décembre

Vos devoirs chaque semaine

- ▶ Visionner les vidéos du MOOC et faire les quizz pour le lundi suivant
- ▶ Vous inscrire si vous le souhaitez à la séance d'appui de la semaine suivante si elle a lieu et si vous en sentez le besoin (délais chaque jeudi midi)

Algorithme ?

🔧 **solution calculatoire**/mécanique/itérative/... à un **problème**

"Spécification d'un schéma de calcul sous forme d'une suite d'opérations élémentaires obéissant à un enchaînement déterminé"

[Encyclopedia Universalis]

Algorithme

- ▶ suite finie de règles à appliquer,
- ▶ dans un ordre déterminé,
- ▶ à un nombre fini de données,
- ▶ se terminant (i.e., arriver, en un nombre fini d'étapes, à un résultat, et ce quelque soit les données traitées).

Qu'est-ce que la programmation ?



«**PROGRAMME**» :

Conception :
quelles notes en-
chaîner ?

Réalisation : percer
les trous aux bons
endroits

Exécution : tourner la
manivelle

Résultat : mélodie

Algorithme

Un algorithme est un moyen pour un humain de représenter la résolution par calcul d'un problème à une autre personne physique (humain) ou virtuelle (calculateur) ; un algorithme est :

- ▶ séquentiel si ses opérations s'exécutent en séquence,
- ▶ parallèle si certaines de ses opérations s'exécutent en parallèle,
- ▶ réparti si certaines de ses opérations s'exécutent sur plusieurs machines.

Toute une théorie

Formalisation : dans les années (19)30 par des mathématiciens :

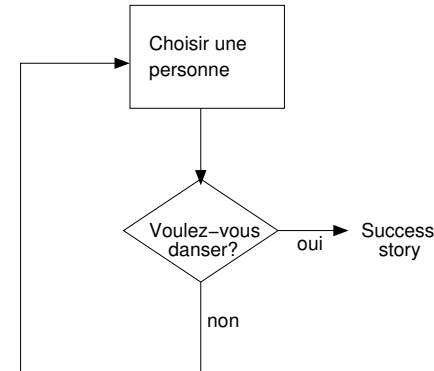
Gödel, Turing, Church, Post, ...

☞ fonctions "calculables" et machines de Turing : abstraction mathématique des notions de **traitement** (suite d'opérations élémentaires), de **problème** et d'**algorithme**.

☞ la partie théorique du cours a pour but de vous en donner un aperçu !

Algorithmes et données

« Voulez-vous danser ? » : premier algorithme :



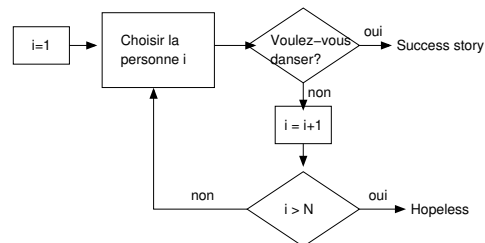
Données :

- ▶ Personne
- ▶ Ensemble de N personnes

☞ Il n'est pas garanti que l'algorithme puisse se terminer !

Algorithmes et données

« Voulez-vous danser ? » : deuxième algorithme :



Données :

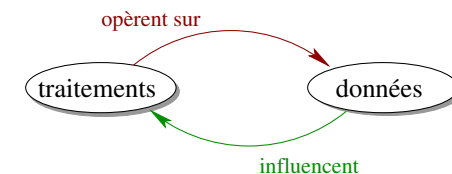
- ▶ Personne
- ▶ Ensemble **ordonné** de N personnes

- ▶ les données sont **structurées**
- ▶ l'algorithme se **termine** nécessairement (au pire N essais successifs)
- ▶ ... mais il n'est pas sûr qu'il soit le plus **efficace** possible !

Conception d'un programme

Concrètement, concevoir un programme c'est **décomposer** la **tâche** à automatiser sous la forme :

- ▶ d'une **séquence d'instructions (traitements)**
- ▶ et de **données** permettant une mise en oeuvre efficace et correcte des traitements.



Formalisation des **traitements** : **algorithmes**

☞ distinguer formellement les bons traitements des mauvais

Formalisation des **données** : **structures de données (abstraites)**

☞ distinguer formellement les bonnes structures de données des mauvaises

Algorithmes – objectifs

On attend d'un algorithme qu'il :

- ▶ se termine,
 - ▶ produise un résultat correct,
 - ▶ pour toute donnée d'entrée valable.
- ☞ Il n'y a (mal)heureusement **aucune recette** pour produire un algorithme !

Algorithme $\neq$ Programme

Un algorithme est **indépendant du langage de programmation** dans lequel on va l'exprimer **et de l'ordinateur** utilisé pour l'exécuter

C'est une description abstraite des étapes conduisant à la solution d'un problème.

Algorithme = partie conceptuelle d'un **programme** (indépendante du langage)

Programme = implémentation (i.e., réalisation) de l'**algorithme**, dans un langage de programmation et sur un système particulier.

- ☞ Premier programme au cours prochain

Difficulté de l'Informatique

Un programme doit être valable pour toute une gamme d'entrées !

- ☞ Impossible de vérifier par des essais : on ne pourra jamais tester tous les cas.

Vérification par preuves mathématiques.

Importance du travail soigneux et mûrement réfléchi !

Lien entre les parties théorique et pratique

- ▶ L'écriture de programmes est indissociable de la conception d'algorithmes.
- ▶ Certains exercices de la partie pratique vous permettront d'expérimenter plus concrètement des problématiques conceptuellement abordées dans la partie théorique :
 - ▶ Ecriture d'algorithmes simples
 - ▶ Mise en pratique de techniques de résolution algorithmiques (récursion, dichotomie)
 - ▶ Théorie de la complexité