

Principes de Programmation

Syllabus

3 janvier 2019

1 Informations

Nom du cours : Principes de Programmation

Niveau : L3

Formation : Licence d'informatique de l'institut Galilée

Page du cours : https://lipn.univ-paris13.fr/~breuvert/principes_de_programmation/

Responsable du cours : Flavien Breuvert

Chargés de TD/TPs : Flavien Breuvert, Joseph Leroux, Rado Rakotonarivo

Email : breuvert@lipn.fr

Bureau : A209

Période du cours : Janvier-Avril 2019

Découpage du cours : 13.5h de CM, 7.5h de TD, 19.5h de TP et 6 phases de projet.

2 Prérequis

Programmation fonctionnel : Il ne s'agit pas d'une initiation à la programmation fonctionnelle, qui a déjà été faite en L2, mais d'un cours avancé se focalisant sur les aspects typage et ordre supérieur. Néanmoins, une partie de la classe arrivant d'une autre formation, les bases de la programmation fonctionnelle seront revu en accélérer. Les élèves concernés sont encouragés à redoubler de travail pour ce cours (en trouvant le temps sur les matières qu'ils ont déjà pu faire dans leur formation initiale).

Automates : Bien qu'elle ne fassent pas partie du cours, la structure d'automate sera abondamment utilisée pour des exercices (y compris des l'examen), et ce sous différents aspects (déterministe, non déterministe, fini, infini...). Une meilleure compréhension des automate vous permettra de vous concentrer sur la difficulté programmatique. Certaines extensions des automates (à pile, à compteur, transducteurs...) seront aussi abordées, mais il ne sera supposé aucune connaissance dessus.

Liste, arbres et graphes : Il est impératif de savoir ce qu'est une liste chaînée et un arbre binaire. Pour le reste, des structures d'arbres et de graphes potentiellement complexe peuvent être utiliser. Il sera attendu que vous sachiez parcourir un arbre, manipuler un graphe et appliquer un algorithme donné dessus.

Java : On n'utilisera pas de Java, mais une compréhension de l'héritage et des interfaces est supposé car elle sera utilisé pendant le cours par comparaison avec les features de Haskell (le langage du cours).

Mathématique : Vous êtes supposés savoir faire une récurrence sur des entiers, mais aussi sur une liste (taille de la liste), ainsi que manipuler des équations par réécriture (simplifications) et manipuler des quantificateurs correctement.

3 Objectifs

Utiliser Haskell : Haskell est le langage de programmation qui sera vu en cours. Bien qu'il ne soit pas particulièrement rependu dans l'industrie, il est souvent considéré comme l'état de l'art de la programmation fonctionnelle : il est toujours mieux d'apprendre une feature fonctionnelle complexe sur Haskell pour pouvoir l'utiliser ailleurs. On en verra la syntaxe, la sémantique, ainsi que certaines features : l'ordre supérieur, le pattern-matching, le polymorphisme, les types-classes...

Polymorphisme et ordre supérieur : On va décrire en détails les notions de typage statique en programmation fonctionnelle à travers le polymorphisme et les type-classes. Il sera attendu que vous compreniez ce qu'est un type de fonction, le polymorphisme et les type-classes et que vous sachiez les utiliser efficacement.

Équivalence de programmes et factorisation : On va définir une notion syntaxique d'équivalence sur les programme et on va raisonner dessus comme on fait des raisonnement algébrique. Il sera attendu que vous arriviez à prouver l'équivalence entre deux programmes pas trop compliqués, potentiellement en utilisant une récurrence sur les structures de données impliqués. Il sera aussi attendu que vous sachiez factoriser un programme en un programme équivalent et plus polymorphe.

Monades : La dernière partie du cours se concentrera sur les monades. Il s'agit d'un sujet complexe qui n'est pas forcément à la portée de tous, mais qui devient de plus en plus essentiel dans de nombreux langages (notamment Scala). Utiliser correctement les monades ne sera pas obligatoire pour valider ce cours, mais le sera pour avoir une bonne note (il y aura aussi des points de cours dessus ;).

Objectifs secondaire : À travers ce cours, vous rencontrerez d'autres notions nouvelles, qu'il ne vous sera pas demandé de pleinement maîtriser, mais il est bon de les rencontrer et d'essayer de les comprendre : l'abstraction, les commentaires, les structures d'erreurs, les benchmarks, certaines structures de données (ABR, automates, arbres préfixe, blockchains...).

Mini-Objectifs du projet : Chaque phase du projet sera associée à un mini-objectif, généralement en lien avec les objectifs du cours.

4 Plan de cours

À venir.

5 Projet

Voir document dédié.

6 TPs

Il y aura alternance entre TPs courts et TPs longs.

TPs courts : D'une durée de 90mn, il s'agira normalement de tutoriels sur les nouveautés vues en cours. Il y aura donc beaucoup de copier-coller et de petites réflexions sur le comportement de telle ou telle commande. J'essaierais de joindre un autre petit exercice optionnel à la fin pour les plus rapides.

TPs longs : D'une durée de 3h, on y verra des exercices plus longs et plus complexes, du type de ce que l'on pourra retrouver à l'examen. Chaque sujet est BEAUCOUP plus long que ce qui est faisable, y sera indiqué le point jusqu'auquel vous êtes sensé aller au minimum, la suite peut être utilisée comme exercice supplémentaire pour réviser (l'un des exercices de l'examen sera repris d'un des suppléments de TP long...).

Correction : Les TPs courts comme les longs ont leur corrigé, mais les exercices supplémentaires des TPs longs n'en ont pas forcément.

7 TDs

Les TDs se focaliseront sur les notions d'équivalence de programme, de factorisation de programme et de type-classes. Ces deux premiers points ne seront pas abordés en TPs et vus assez rapidement en cours, donc ne manquez pas les TDs!

Les corrections des TDs seront disponibles en ligne.

8 Comportement

Retard : Pour les CMs, les retards sont tolérés, mais les retardataire doivent rester silencieux (pas de check!) et s'asseoir derrière. Pour les TP/TDs, cela dépendra du chargé de TP.

Absence : Il est fortement déconseiller de sécher les CM, les TPs ou les TDs. En effet, les TD/TPs ne seront pas des rappel des cours magistraux et seront plutôt des complément. Les années passées, les élèves qui sont venus ponctuellement ont rapidement décrochés, tandis que les élèves présent jusqu'à la fin ont tous validés!¹

Questions par mail : Je répond généralement assez rapidement aux questions par mail (surtout si elles sont courtoises). Si vous n'avez pas de réponses au bout de 3-4 jours, n'hésitez pas à la reposer, je l'ai peut être laissé filée par erreur (je peux avoir jusque 50-60 mails par jours...). Par contre, plus la date de l'examen approche, moins je suis enclin à répondre aux questions (simplement car j'en ai beaucoup plus...).

Lettre de recommandations, recherche de stage et autre : Étant le plus jeune de vos professeurs, je suis facilement sollicité pour différents service académique hors du cours. Si ça peut aider la communication, j'en suis ravis, et j'essaie de satisfaire un maximum de demandes.

1. à une exception près, sachant qu'il a validé le rattrapage.