



Département d'informatique

IFT 359 Programmation fonctionnelle

Plan de cours

Automne 2018

Professeur

Froduald Kabanza

Courriel : kabanza@usherbrooke.ca
 Local : D4-1022-2
 Téléphone : (819) 821-8000, 62865
 Site : planiart.usherbrooke.ca/cours/ift615
 Disponibilité : Vendredi 11 h 30 à 12 h 20 ou sur rendez-vous

Auxiliaires

Frédéric Branchaud-Charron : Frederic.Branchaud-Charron@USherbrooke.ca

Steve Jeffrey Tueno Fotso : Steve.Jeffrey.Tueno.Fotso@USherbrooke.ca

Laurence Fournier : Laurence.Fournier@USherbrooke.ca

Horaire

	Cours			Laboratoire		
Groupe 1	Vendredi	8 h 30 à 10 h 20	Salle D7-2023	Mardi	17h30 D4-1017	
Groupe 2	Vendredi	13 h 30 à 16 h 20	Salle D4-2018	Mardi	17h30 D4-1023	

Note : Parfois les cours ne pourront avoir lieu le vendredi. Des cours de rattrapage seront programmés à une autre période.

Description officielle de l'activité pédagogique¹

Objectifs	Formaliser les notions d'abstraction procédurale et d'abstraction de données dans le cadre de la programmation fonctionnelle.
Contenu	Qualité, modularité, conception fonctionnelle. Processus récursifs et itératifs. Objets atomiques. Listes. Abstraction d'ordre supérieur. Curryfication. Fermeture. Appels terminaux. Modèle d'exécution d'un programme fonctionnel. Application de la programmation fonctionnelle (structure de donnée non mutable, programmation par flots, <i>pattern matching</i> ...). Insistance sur la qualité de la solution.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure de laboratoire par semaine 5 heures de travail personnel par semaine
Préalable	IFT 159

¹ <https://www.usherbrooke.ca/admission/fiches-cours/IFT359/>

1 Présentation

Cette section présente les cibles de formation et le contenu détaillé de l'activité pédagogique.

1.1 Mise en contexte

Le cours IFT 359 introduit les étudiants à la programmation fonctionnelle. La *programmation fonctionnelle* est un paradigme de programmation selon lequel les algorithmes sont vus fondamentalement comme des fonctions mathématiques. La notion d'état d'un programme n'est pas explicite dans le style de programmation. Elle a comme base théorique le *lambda calcul*. Pour comparaison, le paradigme de *programmation impérative* dans les langages comme C suit un style selon lequel un état d'un programme change en fonction d'opérations effectuées sur l'état. La base théorique sous-jacente est la *machine de Turing*. Le paradigme de programmation orienté-objet est basé sur le concept d'objet : un objet peut contenir des données et peut avoir un comportement défini par des méthodes. Le paradigme de programmation logique est basé sur l'inférence déductive avec le calcul des prédicats.

Plusieurs langages de programmation populaires supportent le paradigme de programmation fonctionnel à des degrés divers : C#, Python, Java, C++ et d'autres. La programmation fonctionnelle est prépondérante dans JavaScript et Scala. Elle est l'essence même des langages Haskell, Racket, Racket, Scheme et Lisp. En fait, Lisp est le tout premier langage de programmation fonctionnel, créé en 1958 par John McCarthy. Jusqu'au début des années 1990, c'était le langage de prédilection pour les recherches en intelligence artificielle.

Étant donné l'influence que le paradigme de programmation fonctionnelle a eu et continuera probablement d'avoir sur les langages de programmation, il est intéressant d'en connaître les principes et les pratiques sous-jacents.

1.2 Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant devrait :

1. Connaître les principes et les pratiques sous-jacents à la programmation fonctionnelle : abstractions de données, abstractions de fonctions, évaluation des fonctions en ordre normal et en ordre applicatif, transparence référentielle, fonction d'ordre supérieur, curryfication.
2. Savoir implémenter un algorithme dans un langage fonctionnel choisi : définition, utilisation et création des identificateurs, des fonctions et des formes syntaxiques (instructions de contrôle), des structures de données basiques (liste, paire, etc.)
3. Savoir utiliser les notions clés de la programmation fonctionnelle (récursivité, fonction d'ordre supérieur, fermeture, récursivité terminale, etc.).
4. Pouvoir expliquer le déroulement de l'exécution d'un programme fonctionnel : évaluation par environnement d'un programme fonctionnel.
5. Appliquer des techniques avancées de programmation en s'appuyant sur les propriétés des langages fonctionnels (p. ex. évaluation paresseuse, création de nouvelles formes syntaxiques, création d'un langage embarqué (p. ex. langage à objet), pattern matching).

1.3 Contenu détaillé

Thème	Cours Magistraux	Heures	Objectifs
1	Principes théoriques : lambda-calcul ; fonction d'ordre supérieur ; évaluation en ordre normal ; évaluation en ordre applicatif ; transparence référentielle ; curryfication	6	1
2	Introduction au langage fonctionnel choisi : définition des identificateurs ; définition des fonctions ; portée lexicale ; conditions et opérateurs de logique ; instructions de contrôle ;	5	2

	typage ; structure de données basiques (liste, paire, etc.)		
3	Évaluation par environnement d'un programme	4	4
4	Fonctions récursives	4	3, 4
5	Fonctions d'ordre supérieur	4	3, 4
6	Appel terminaux	4	3, 4
7	Fermeture, affectation ou monade	3	3, 4, 5
5 activités d'approfondissement parmi les 8 suivantes			
8	Création de nouvelles formes syntaxiques	4	3, 5
9	Structure de données non mutable.	4	3, 5
10	Méta-évaluateur circulaire en ordre applicatif Évaluation par environnement	4	3, 5
11	Méta-évaluateur circulaire en ordre normal ; programmation par flots; évaluation paresseuse.	4	3, 5
12	Création de langage embarqué (p. ex. création d'un langage objet).	4	3, 5
13	Utilisation explicite de continuation	4	3, 5
14	Études des types : construction et preuves de programme.	4	3, 5
15	Appariement de forme.	4	3, 5

Le nombre d'heures inclut les exercices dirigés. Le cours comprend au moins quatre travaux pratiques qui couvrent les thèmes choisis.

2 Organisation

Cette section présente la méthode pédagogique utilisée, le calendrier officiel du cours, la méthode d'évaluation ainsi que l'échéancier des travaux.

2.1 Méthode pédagogique

Le cours se donnera sous forme de trois heures d'exposé magistral et une heure de laboratoire par semaine. Cinq devoirs permettront aux étudiants de mettre en pratique la matière couverte.

2.2 Contenu détaillé et calendrier

Les sections font référence au livre [Structure and Interpretation of Computer Programs \(SICP\), Abelson et al., 1996](#). Le livre est basé sur le langage Scheme, alors que le cours utilise Racket. Il y a quelques différences minimales entre les deux, qui seront expliquées progressivement, sans que cela nuise à la compréhension des concepts fondamentaux. Les dates dans le calendrier suivant pourraient varier.

Date	Cours Magistraux	Sections dans le livre SICP
	Présentation du plan d'activité pédagogique	
VE 7/9	Abstraction avec les fonctions élémentaires, Partie 1 (Racket): <ul style="list-style-type: none"> Éléments de base : Expressions arithmétiques, définition de variables et de fonctions, évaluation en ordre applicatif vs évaluation en ordre normal, fonctions internes, procédures récursives 	1.1
VE 14/9	Abstraction avec les fonctions élémentaires, Partie 2 (Racket) : <ul style="list-style-type: none"> Récursivité terminale (<i>tail recursion</i>) 	1.2
ME 19/9	Abstractions avec des fonctions d'ordre supérieur (Racket) : <ul style="list-style-type: none"> Fonctions comme arguments Expressions <i>lambda</i> Fonctions retournées comme valeur Curryfication 	1.3
VE 21/9	Abstraction avec des données, Partie 1 (Racket) : <ul style="list-style-type: none"> Structures de données : Paires, listes, ensembles, vecteurs, tables hash (map), chaînes de caractères 	2.1
VE 28/9	Abstraction avec des données, Partie 2 (Racket) : <ul style="list-style-type: none"> Affectation Données mutables 	
VE 5/10	Révisions en vue de l'intra	
Examens périodiques du 6 au 13 octobre (pas de cours)		
Relâche du 15 au 19 octobre (pas de cours)		

	Correction de l'intra	
VE 26/10	Fermeture (<i>closure</i>) (Racket)	2.2
ME 31/10	Modularité : Consolidation des concepts vus jusqu'ici en suivant un exemple orienté-objet (Racket)	2.4
VE 2/11	Évaluation par environnement d'un programme	3.2
VE 9/11	Programmation fonctionnelle en Python, Partie 1	
VE 16/11	Pas de cours	
VE 23/11	Programmation fonctionnelle en Python, Partie 2	
VE 30/11	Programmation fonctionnelle en Python, Partie 3	
VE 7/11	Révisions en vue de l'examen final	
Examens finaux du 11 au 21 décembre (pas de cours)		
Fin du cours		

2.3 Évaluation

Travaux pratiques :	50 %
Examen périodique :	15 %
Examen final :	35 %

Les examens sont à livre fermé. Vous avez droit à une feuille d'aide-mémoire personnelle, recto verso.

2.4 Qualité du français et de la présentation

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages², l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3³, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à moins que leur usage ne soit explicitement permis dans le plan de cours.

Dans ce cours, l'usage de téléphones cellulaires est interdit. Celui des de tablettes ou d'ordinateurs est autorisé uniquement pour suivre le cours. Le son doit être coupé en tout temps. Cette permission peut être retirée en tout temps si leur usage entraîne des abus.

Comme indiqué dans le règlement universitaire des études, section 4.2.3⁴, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission du professeur.

² https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/Etudiants_actuels/Informations_academiques_et_reglements/2017-10-27_Reglement_facultaire_-_evaluation_des_apprentissages.pdf

³ https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Intranet/Informations_academiques/Sciences_Reglement_complementaire_2017-05-09.pdf

⁴ https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Intranet/Informations_academiques/Sciences_Reglement_complementaire_2017-05-09.pdf

2.6 Échéancier et critères d'évaluation du projet

Travail pratique	Publication de l'énoncé	Date limite de remise	Pondération
TP 1	Vendredi 14 sep	Vendredi 21 sep à minuit	10 %
TP 2	Vendredi 21 sep	Vendredi 28 sep à minuit	10 %
TP 3	Vendredi 2 nov	Vendredi 9 nov à minuit	10 %
TP 4	Vendredi 9 nov	Vendredi 16 nov à minuit	10 %
TP 5	Mercredi 16 nov	Vendredi 7 dec à minuit	10 %

Les énoncés, les critères d'évaluation et les autres directives seront communiqués au moment approprié. Les directives particulières suivantes vont s'appliquer.

- Toute soumission d'un travail pratique en retard vaudra zéro, à moins qu'elle soit motivée par des raisons valables et conformes au règlement des études (par exemple, maladie avec attestation du médecin).
- Les devoirs pourront se faire par groupe de deux.

3 Matériel pour le cours

- Une bonne partie du cours est basée sur le livre [Structure and Interpretation of Computer Programs \(SICP\)](#), [Abelson et al., 1996](#).
- L'environnement de développement et la documentation de Racket, [disponibles en ligne](#), sont essentiels.
- De la documentation spécifique à la programmation fonctionnelle en Python sera fournie en temps et lieu.



L'intégrité intellectuelle passe, notamment, par la reconnaissance des sources utilisées. À l'Université de Sherbrooke, on y veille!

Extrait du Règlement des études (Règlement 2575-009)

9.4.1 DÉLITS RELATIFS AUX ÉTUDES

Un délit relatif aux études désigne tout acte trompeur ou toute tentative de commettre un tel acte, quant au rendement scolaire ou une exigence relative à une activité pédagogique, à un programme ou à un parcours libre.

Sont notamment considérés comme un délit relatif aux études les faits suivants :

- a) commettre un plagiat, soit faire passer ou tenter de faire passer pour sien, dans une production évaluée, le travail d'une autre personne ou des passages ou des idées tirés de l'œuvre d'autrui (ce qui inclut notamment le fait de ne pas indiquer la source d'une production, d'un passage ou d'une idée tirée de l'œuvre d'autrui);
 - b) commettre un autoplagiat, soit soumettre, sans autorisation préalable, une même production, en tout ou en partie, à plus d'une activité pédagogique ou dans une même activité pédagogique (notamment en cas de reprise);
 - c) usurper l'identité d'une autre personne ou procéder à une substitution de personne lors d'une production évaluée ou de toute autre prestation obligatoire;
 - d) fournir ou obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour une production faisant l'objet d'une évaluation;
 - e) obtenir par vol ou toute autre manœuvre frauduleuse, posséder ou utiliser du matériel de toute forme (incluant le numérique) non autorisé avant ou pendant une production faisant l'objet d'une évaluation;
 - f) copier, contrefaire ou falsifier un document pour l'évaluation d'une activité pédagogique;
- [...]

Par plagiat, on entend notamment :

- Copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets;
- reproduire des présentations, des dessins, des photographies, des graphiques, des données... sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire;
- utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources;
- résumer ou paraphraser l'idée d'un auteur sans en indiquer la source;
- traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets ;
- utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord);
- acheter un travail sur le Web ou ailleurs et le faire passer pour sien;
- utiliser sans autorisation le même travail pour deux activités différentes (autoplagiat).

Autrement dit : mentionnez vos sources