

Université de
Sherbrooke

Département d'informatique

IFT 359 – Programmation fonctionnelle

Plan d'activité pédagogique

Automne 2021

Enseignant

Hélène Pigot

Courriel : helene.pigot@usherbrooke.ca

Local : D4-2002

Téléphone : +1 819 821-8000 x63078

Disponibilités : Consultation : jeudi de 10 h 30 à 12 h dans le bureau de la professeure (D4-2002) ou sur Teams. De préférence prendre rendez-vous préalablement.

Responsable(s) : Direction du département

Site web du cours : <https://moodle.usherbrooke.ca>

Horaire

Groupe 1 :	Exposé magistral :	Jeudi	15h30 à 17h20	salle D7-2023
		Vendredi	13h30 à 15h20	salle D3-2035
	Exercices/laboratoires :	Jeudi	15h30 à 17h20	salle D4-1023/1017
Groupe 2 :	Exposé magistral :	Jeudi	13h30 à 15h20	salle D7-2023
		Vendredi	8h30 à 10h20	salle D7-2021
	Exercices/laboratoires :	Jeudi	13h30 à 15h20	salle D4-1023/1017

Description officielle de l'activité pédagogique¹

Cibles de formation :	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, pattern matching...). Insistance sur la qualité de la solution.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure d'exercices par semaine 5 heures de travail personnel par semaine
Préalable	IFT 159
Particularités	Aucune

¹<https://www.usherbrooke.ca/admission/fiches-cours/ift359>

1 Présentation

Cette section présente les cibles de formation spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation du comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

Le cours IFT 359 introduit les étudiantes et les étudiants à la programmation fonctionnelle. La programmation fonctionnelle a été introduite avec le langage Lisp créé par John McCarthy en 1958, dans les mêmes années où les langages de haut niveau supportant la programmation impérative (procédurale/structurée) tels que Fortran (1957) et Algol (1958) en étaient à leurs débuts. Lisp est le tout premier langage de programmation fonctionnel et jusqu'au début des années 1990, c'était le langage de prédilection pour les recherches en intelligence artificielle.

La programmation fonctionnelle se distingue des autres styles de programmation (impérative, logique, objets) par le fait que les algorithmes sont vus fondamentalement comme des fonctions mathématiques. La programmation fonctionnelle est basée sur un modèle mathématique (lambda calcul) dans lequel une fonction, souvent récursive, prend une entrée et calcule une sortie. La notion d'état d'un programme n'est pas explicite dans ce style de programmation.

À l'opposé du fonctionnel, la programmation impérative, utilisée dans les langages tels C++ et Java, est basée sur un modèle où un état change en fonction des opérations effectuées. La notion d'état d'un programme est très présente dans ce style de programmation. La base théorique sous-jacente est la machine de Turing qui est très proche de l'implémentation physique (registres, mémoire, jeu d'instructions de l'UCT, etc.).

De même, le paradigme de programmation orienté-objet, qui est basé sur le concept d'objet pouvant contenir des données et des méthodes, reste une forme de programmation impérative dans laquelle la notion d'état demeure très présente.

Finalement, l'approche fonctionnelle se distingue aussi de la programmation logique car cette dernière est basée sur l'inférence déductive avec le calcul des prédicats. La programmation fonctionnelle est restée dans l'ombre de la programmation orientée objet, mais elle refait maintenant surface, car ses propriétés sont très utiles pour faire face à certains problèmes actuels (e.g. LINQ, MapReduce).

De nombreux langages utilisent ou sont basés sur l'approche fonctionnelle dont :

1. Le langage Erlang (1986), développé par Ericsson, qui se base sur la programmation fonctionnelle pour offrir un langage propre à la programmation concurrente, parallèle et temps réel ;
 2. Le langage Scheme et ses dialectes (Racket, MIT Scheme), souvent utilisés dans le milieu académique, qui est un langage très bien supporté par de nombreux outils et des compilateurs efficaces ;
 3. Le langage Haskell (1990 et 1998) qui a été développé pour consolider les acquis des dernières décennies et qui se démarque par son typage statique totalement inféré, ses effets de bord contrôlés et son évaluation paresseuse ;
 4. Le langage JavaScript (1995) qui supporte de nombreux concepts de programmation fonctionnelle ;
 5. le langage Autolisp (1986) qui est fréquemment utilisé par les ingénieurs à l'intérieur d'Autocad ;
 6. Le EmacsLisp (1976) permet d'adapter l'éditeur Emacs à un grand nombre d'applications ou langage informatique ;
 7. les langages Common Lisp (1984), OCaml (1996), Scala (2003) et F# (2008) qui combinent la programmation fonctionnelle et la programmation orientée objet ;
 8. les langages C#, Python, Java, C++ qui intègrent maintenant plusieurs notions de programmation fonctionnelles.
- Étant donné l'influence que le paradigme de programmation fonctionnelle a eue 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 Cibles de formation 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, lambda-calcul, curryfication ;

2. Savoir implémenter un algorithme dans un langage fonctionnel choisi : définition, utilisation et création des identificateurs et des fonctions ; formes syntaxiques (opérateurs logiques, instructions de contrôle) ; portée lexicale, 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	Contenu	Nbr. d'heures	Objectifs	Travaux	Lectures
1	Principes théoriques : lambda-calcul, fonction d'ordre supérieur, évaluation en ordre normal, évaluation en ordre applicatif, transparence référentielle, curryfication.	6	1	✓	[1] section 1.3.2 [2] chapitre 20, p. 591-626
2	Introduction au langage fonctionnel choisi : définition des identificateurs, définition des fonctions, portée lexicale, conditions et opérateurs de logique, instructions de contrôle, typage, structure de données basiques (liste, paire, etc.).	5	2	✓	[1] section 1.1 [2] chapitre 1, p. 3-23 ; chapitre 2, p. 25-58
3	Évaluation par environnement d'un programme : évaluation par environnement d'un programme.	4	4	✓	[1] section 3.2 [2] sections 2.5 et 2.6, p. 40-46
4	Fonctions récursives : fonctions récursives.	4	3 et 4	✓	[1] section 1.2 [2] section 1.7, p. 17-21 ; section 2.5, p. 40-45
5	Fonctions d'ordre supérieur : fonctions d'ordre supérieur.	4	3 et 4	✓	[1] sections 1.3 et 2.2 [2] sections 3.1 et 3.6, p. 65-85
6	Appels terminaux : appels terminaux.	4	3 et 4	✓	[1] sections 1.2.1 à 1.2.4 [2] sections 3.7 à 3.11, p. 83-99
7	Fermeture, affectation ou monade : fermeture, affectation ou monade.	3	3, 4 et 5	✓	[1] sections 3.1 et 3.3 [2] chapitre 5, p. 125-159
8	Création de nouvelles formes syntaxiques : création de nouvelles formes syntaxiques.	4	3 et 5	✓	[1] sections 4.1.2 à 4.1.7 [2] chapitre 9, p. 269-304
9	Structure de données non mutable : structure de données non mutable.	4	3 et 5	✓	[1] chapitre 2 [2] chapitre 7, p. 185-234
10	Évaluation : méta-évaluateur circulaire en ordre applicatif, évaluation par environnement.	4	3 et 5		[1] sections 3.5 et 4.1 [2] sections 10.1 à 10.4, p. 305-317

11	Évaluation (suite) : méta-évaluateur circulaire en ordre normal, programmation par flots, évaluation paresseuse.	4	3 et 5	✓	[1] sections 3.5 et 4.2 [2] sections 10.5 à 10.9, p. 313-319
12	Création de langage embarqué : e.g. création d'un langage objet.	4	3 et 5		[2] chapitre 11, p. 331-378
13	Utilisation explicite de continuation : utilisation explicite de continuation.	4	3 et 5		[2] chapitre 8, p. 235-268
14	Études des types : construction et preuves de programme.	4	3 et 5		[2] chapitre 16, p. 491-514
15	Appariement de forme : appariement de forme.	4	3 et 5		[1] section 4.4 [2] chapitre 17, p. 515-536

1. Le cours doit comprendre au moins cinq travaux pratiques couvrant tous les sujets marqués «✓» dans le tableau.
2. Les lectures indiquées ne sont là qu'à titre indicatif. L'enseignant est libre de choisir un autre document de référence.

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Des exposés théoriques magistraux auront lieu le vendredi. Ils seront ensuite appliqués dans les laboratoires le jeudi suivant. Il est attendu que les étudiants et étudiantes adoptent le paradigme fonctionnel. Cela demande beaucoup de pratique et un apprentissage actif pour solutionner les exercices proposés en cours et en laboratoire. Des exercices corrigés seront proposés pour les exercices des laboratoires les plus importants.

Compte tenu du contexte actuel (pandémie due au COVID-19), il se peut que le cours ait lieu en totalité ou en partie à distance d'une façon différente de ce qui est énoncé ci-dessus. Notez que vous en serez informés rapidement si tel est le cas.

2.2 Calendrier

Semaine	Date	Thème	Lectures
1	2021-08-30	1 et 2	[1] 1.2; [2] chap1 p1-35
2	2021-09-06	2 et 4	[1] 1.1.5; 1.2; 2.5
3	2021-09-13	2, 4 et 5	[1] 1.3; [2] 3.1 à 3.4
4	2021-09-20	2, 4 et 5	[1] 2.1; [2] chap2 2.1 à 2.8
5	2021-09-27	2, 5 et 6	[1] 2.2; [2] 7.8 à 7.10
6	2021-10-04	5 et 6	[1] 1.3; [2] 3.4 à 3.6
7	2021-10-11	3 et 10	[1] 1.3; [2] 3.4 à 3.6
8	2021-10-18	Examen périodique	
9	2021-10-25	Relâche	
10	2021-11-01	9 et 11	[1] 2.2; [2] 3.6; 7.8 à 7.10
11	2021-11-08	11	[1] 3.5; [2] 10.4 à 10.7
12	2021-11-15	3 et 7	[1] 3.1 à 3.3; [2] 5.1, 5.2, 5.5 à 5.8
13	2021-11-22	8	[2] chap 9; Aide Dr Racket :chap 16
14	2021-11-29	13	[2] chap 8
15	2021-12-06	Révision	
16	2021-12-13	Examen final	

La première semaine (semaine du 31 août) le cours du lundi aura lieu en laboratoire et il n'y aura pas de laboratoire cette semaine là ni le jeudi ni le vendredi.

2.3 Évaluation

Devoir (5)	35 %
Examen intra	25 %
Examen final	40 %

En raison de la situation particulière de la COVID qui, d'une part modifie les méthodes pédagogiques et, d'autre part peut provoquer des modifications dans la façon de donner le cours, la professeure, en accord avec les étudiantes et étudiants, se réserve le droit de modifier les pondérations des évaluations.

2.3.1 Qualité de la langue et de la présentation

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages² l'enseignante ou 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.

2.3.2 Plagiat

Le plagiat consiste à utiliser des résultats obtenus par d'autres personnes afin de les faire passer pour sien et dans le dessein de tromper l'enseignante ou l'enseignant. Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 9.4.1 du Règlement des études³. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 9.4.1 du Règlement des études de l'Université de Sherbrooke. L'étudiante ou l'étudiant peut s'exposer à de graves sanctions qui peuvent être soit l'attribution de la note E ou de la note zéro (0) pour un travail, un examen ou une activité évaluée, soit de reprendre un travail, un examen ou une activité pédagogique. Tout travail suspecté de plagiat sera transmis au Secrétaire de la Faculté des sciences. Ceci n'indique pas que vous n'avez pas le droit de coopérer entre deux équipes, tant que la rédaction finale des documents et la création du programme restent le fait de votre équipe. En cas de doute de plagiat, l'enseignante ou l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'elle ou qu'il considère comme étant plagié. En cas d'incertitude, ne pas hésiter à demander conseil et assistance à l'enseignante ou l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

Devoir	Sujet	Réception	Remise	Points
TP1	Introduction	À définir	2021-09-15	5
TP2	Récurtivité	À définir	2021-09-29	5
TP3	Fonction d'ordre supérieur	À définir	2021-10-03	5
TP4	Programmation par flot	À définir	2021-11-17	8
TP5	Analyse syntaxique	À définir	2021-12-08	12

2.4.1 Directives particulières

Les travaux pratiques peuvent être faits en équipe de deux personnes (ceux qui travailleront seuls seront évalués avec le même barème). Aucun retard n'est admis sauf pour les mêmes motifs que les absences aux examens (voir le règlement des études). L'évaluation tiendra compte de l'exactitude des résultats, mais aussi de l'utilisation judicieuse des concepts et du respect des standards de programmation. Une erreur de soumission entraînera automatiquement une pénalité d'au moins 10 %.

Les travaux seront remis par Turnin.

Les dates de remise des devoirs et leur sujet sont présentés à titre indicatif. Ils peuvent être sujet à changement.

Les résultats des évaluations, les notes de cours, les directives des travaux, les exercices, les exemples faits avec DrRacket en classe seront placés sur le site Moodle du cours.

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁴, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à condition que leur usage soit explicitement permise dans le plan de cours.

Dans ce cours, l'usage de téléphones cellulaires, de tablettes ou d'ordinateurs est autorisées. Cette permission peut être retirée en tout temps si leur usage entraîne des abus.

Tel qu'indiqué dans le règlement universitaire des études, section 4.2.3⁵, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission de la personne enseignante.

Note : L'utilisation du courriel est recommandée pour poser vos questions.

²https://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/registraire/droits-et-responsabilites/reglement-des-etudes/>

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

⁵<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

3 Matériel nécessaire pour l'activité pédagogique

Une bonne partie du cours est basée sur les deux livres de Abelson et al. [1] et Chazarain [2]. L'acquisition d'un des deux ouvrages est fortement recommandé.

L'environnement de développement et la documentation de Racket, disponibles en ligne, sont essentiels.

4 Références

- [1] ABELSON, HAROLD AND SUSSMAN, GERALD J. : *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, MA, USA, 2nd édition, 1996.
- [2] CHAZARAIN, J. : *Programmer avec Scheme : de la pratique à la théorie*. International Thomson Publ. France, 1996.



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