



Département d'informatique

IFT 159 – Analyse et programmation

Plan d'activité pédagogique Automne 2018

Enseignant

Vincent Ducharme

Courriel :	Vincent.Ducharme2@usherbrooke.ca
Local :	D4-1010-20
Téléphone :	(819) 821-8000 poste 66186
Site :	http://info.usherbrooke.ca/vducharme/ift159
Disponibilité :	Sur rendez-vous

Responsable(s) : Gabriel Girard

Horaire

Groupe 1 :	Exposé magistral :	Lundi	10h30 à 12h20	salle D3-2041
		Vendredi	8h30 à 10h20	salle D3-2038
	Exercices/laboratoires :	Mardi	8h30 à 9h20	salle D4-1017/D4-1023
Groupe 2 :	Exposé magistral :	Lundi	8h30 à 10h20	salle D3-2041
		Vendredi	13h30 à 15h20	salle D3-2038
	Exercices/laboratoires :	Jeudi	13h30 à 14h20	salle D4-1017/D4-1023

Description officielle de l'activité pédagogique ¹

Cibles de formation : Savoir analyser un problème; avoir un haut degré d'exigence quant à la qualité des programmes; pouvoir développer systématiquement des programmes de bonne qualité, dans le cadre de la programmation procédurale séquentielle.

Contenu : Introduction aux ordinateurs. Analyse et conception de solutions informatiques : simplification, décomposition, modularisation et encapsulation. Critères de qualité : la validité, la fiabilité, la modifiabilité et les tests. Concepts de base de la programmation structurée : séquence, itération, sélection. Modélisation du traitement et modularité : concept de fonctions et d'abstraction procédurale. Concept de base de l'abstraction de données. Introduction aux concepts orientés objet : classe, constructeur, surcharge, notation UML (diagramme de classes). Récursivité. Processus logiciel personnel (PSPO).

Crédits 3
Organisation 3 heures d'exposé magistral par semaine
 1 heure d'exercices par semaine
 5 heures de travail personnel par semaine

Particularités Aucune

1. <https://www.usherbrooke.ca/admission/fiches-cours/ift159.html>

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 d'un comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

Le cours IFT 159, Analyse et Programmation, est le premier cours d'informatique des baccalauréats en science de l'image et des médias numériques, en informatique, en informatique de gestion et en mathématiques. Il ne présuppose pas de connaissance en programmation. Il ne requiert que la connaissance de l'utilisation usuelle d'un ordinateur. Comme son nom l'indique c'est un cours d'analyse de problèmes et de programmation. L'analyse consiste à l'ensemble des activités dédiées à l'étude détaillée d'un problème. La programmation comprend des activités de conception, de codage, de test et de maintenance de programmes pour ordinateurs. Dans le cadre du cours IFT 159, nous traitons des notions d'analyse, de conception, de codage et de tests. Nous ne parlons pas de la maintenance.

Tout analyste-programmeur qui se voit confier le développement d'un système doit être capable d'analyser le problème demandé et fournir la conception détaillée de la solution retenue. De plus, l'analyste-programmeur doit être capable de mettre en oeuvre une conception écrite par un autre analyste. Le programmeur doit aussi penser aux différents tests à effectuer sur le système développé afin de s'assurer de son bon fonctionnement. Dans le cadre des travaux pratiques, vous aurez à effectuer chacune des tâches de l'analyste-programmeur.

Ce cours est aussi le premier d'une chaîne de cours d'analyse de problèmes et de programmation: IFT 339 et IFT 232 en deuxième session, IFT 359 et IFT 287 en troisième session, puis éventuellement les cours de génie logiciel (IGL). De plus, les connaissances acquises en programmation dans ce cours sont primordiales pour les cours de IFT 209, IFT 215, IFT 232 et IFT 287.

1.2 Cibles de formation spécifiques

L'objectif du cours est d'apprendre à résoudre un problème en utilisant l'informatique. Le langage utilisé dans le cours est C++. Il est à noter que le but du cours n'est pas d'apprendre un langage mais bien le processus menant à résoudre un problème en utilisant l'informatique ainsi que les concepts généraux utilisés par les langages de programmation.

De façon plus précise, à la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable :

1. de comprendre le cycle de développement d'une application simple;
2. d'utiliser des outils de développement logiciel;
3. de lire et comprendre un énoncé de problème peu complexe et procéder à son analyse;
4. d'analyser un problème pour y offrir une solution algorithmique;
5. d'illustrer un algorithme en utilisant la représentation appropriée;
6. de mettre en oeuvre un algorithme à l'aide d'un langage de programmation;
7. de planifier des tests pour un programme;
8. de vérifier le bon fonctionnement d'un programme;
9. de respecter des normes et standards de programmation;
10. d'utiliser les mécanismes élémentaires d'encapsulation orientée objet;
11. de rédiger un programme en appliquant les principes de base de la dérivation par enrichissements successifs;
12. de documenter la solution au moyen d'un document séparé (analyse et conception) ou d'une documentation incluse (programmation).

1.3 Contenu détaillé

Thème	Contenu	Heures	Objectifs	Travaux
1	Brève introduction aux ordinateurs : modèle pratique du calcul, modèle théorique du calcul, étapes de la création, de la mise en oeuvre et de l'exécution d'un programme.	3	1	
2	Développement de programmes : cycle de vie et phases de développement du logiciel; éléments de base du C++, notions de programmes (variables, constantes, types, énoncés, fonctions, ...), notions de compilation, notions de gestion d'effort (PSP0).	4	2..8,10,11	A/P
3	Analyse et conception descendante : phases de développement, spécification, analyse, conception, programmation, tests fonctionnels; introduction aux fonctions.	5	2..8,10,11	A/P
4	Structures sélectives des langages : expressions logiques; énoncé « if », énoncés composés, énoncé « if » emboîté, énoncé « switch » ; notions d'analyse et de conception.	5	2..8,10,11	A/P
5	Structures itératives des langages : concept d'itération; boucle conditionnelle, de parcours et de comptage; analyse et conception; énoncés « while », « for » et « do ... while » ; récursivité; boucles emboîtées.	5	2..8,10,11	A/P
6	Les fonctions et introduction à la récursivité : concept de modularité; fiabilité des fonctions (validité, robustesse, assertions); utilisations avancées des fonctions; expressions logiques; récursivité; paramètres de sortie; tests unitaires et systèmes; analyse et conception.	5	2..8,10,11	A/P
7	Organisation des données et types : représentation interne des données; création de nouveaux types simples; types énumérés; tableaux : concept et utilité des tableaux, tableaux à une dimension, tableaux à plusieurs dimensions, passage de tableaux en paramètre; types structurés (enregistrements); tableaux d'enregistrements; ensembles; utilisation d'une bibliothèque («vector»).	7	2..8,10,11	A/P
8	Introduction à l'abstraction de données : encapsulation; concepts d'abstraction de données; introduction aux classes; introduction à UML : diagramme de classes.	8	9	A/P
9	Récursivité : caractéristiques d'un problème récursif; exemples de problèmes récursifs.	3	3..5	
10	Introduction à la complexité algorithmique : définition; concept d'ordre de complexité; calcul de complexité simple, exemples.	2	3	
11	Conclusion.	1		

1. Les heures associées à un thème particulier inclues les heures d'exercices pour un total de 48 heures (12 semaines de quatre heures).
2. Le cours doit comprendre au moins cinq travaux pratiques couvrant tous les sujets marqués dans le tableau. Les lettres « A » et « P » représentent respectivement l'analyse/conception et la programmation.
3. Pour chaque travail, des séries de tests devront être remis ainsi qu'un estimé et un compte rendu de l'effort requis pour effectuer le travail (PSP0).

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

Une semaine comprend cinq heures de présence en classe: trois heures de cours magistral, une heure d'exercices et une heure en laboratoire.

Chaque semaine, il y aura environ trois heures d'exposés magistraux décrivant la théorie ainsi que des exemples développés au tableau. Il y aura aussi environ une heure d'exercices et d'exemples qui seront directement intégrés dans les cours magistraux. Au moins une étude de cas sera abordée : analyse de la problématique, implémentation de celle-ci. On reviendra alors sur les éléments nouveaux de langage vus dans la programmation de la solution. Il y a une heure d'assistance en laboratoire prévue à l'horaire. Elle servira principalement à faire du soutien et du dépannage technique et vous aider dans la réalisation des laboratoires et des devoirs. C'est aux étudiantes et étudiants que revient la tâche de s'organiser pour pratiquer ce qui sera vu en cours.

2.2 Calendrier

Semaine	Date	Thèmes	Laboratoire	Lecture
1	27 août	1		Chapitre 1-2 de [4]
2	3 sept.	2	Présentation de l'environnement	Chapitre 1-2 de [4]
3	10 sept.	2-3	Environnement et compilation	Chapitre 2-3 de [4]
4	17 sept.	4-5	Séquence et fonctions	Chapitre 3 de [4]
5	24 sept.	5	Sélection/itération	Chapitre 4 de [4]
6	1 oct.	7	Array et vecteurs	Chapitre 4-5 de [4]
7	8 oct.	Examen intra		
8	15 oct.	Relâche		
9	22 oct.	7-8		Chapitre 5-6 de [4]
10	29 oct.	7-8		Chapitre 5-6 de [4]
11	5 nov.	9	Classes	Chapitre 6 et sections 9.1, 9.2, 9.4, 9.5 et 11.1 de [4]
12	12 nov.	6-9	Tableau	Sections 9.3, 9.7 et 9.8 de [4]
13	19 nov.	6	Récursivité	Chapitre 10, sections 9.3, 9.7 et 9.8 de [4]
14	26 nov.	10		Chapitre 10, sections 9.3, 9.7 et 9.8 de [4]
15	3 déc.	11	Librairies	Chapitre 12, section 11.7 de [4]
16	10 déc – 17 déc.	Révision et examen final		

2.3 Évaluation

Tests	Intra	Final	Devoirs	Laboratoires
25 %	5 %	30 %	30 %	10 %

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

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'enseignant. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 9.4.1 du *Règlement des études*³ de l'Université de Sherbrooke. L'étudiant ou l'étudiante peut s'exposer à de graves sanctions dont automatiquement un zéro (0) au devoir ou à l'examen en question.

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 reste le fait de votre équipe. En cas de doute de plagiat, l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'il considère comme étant plagié. En cas de doute, ne pas hésiter à demander conseil et assistance à l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

TP	Réception de l'énoncé	Thème	Remise
1	10/09/18	Premier programme	17/09/18 (à 23h59)
2	17/09/18	Fonctions	24/09/18 (à 23h59)
3	24/09/18	Sélection et itération	01/10/18 (à 23h59)
4	01/10/18	Tableaux et vecteurs	08/10/18 (à 23h59)
5	08/10/18	Analyse/Conception	22/10/18 (à 23h59)
6	22/10/18	Classes	05/11/18 (à 23h59)
7	05/11/18	Réursivité	19/11/18 (à 23h59)
8	19/11/18	Programme complexe	11/12/18 (à 23h59)

Directives particulières :

Ponctualité

Vous devez arriver à l'heure au cours. Une fois la porte fermée, elle n'est réouverte qu'à la pause.

Tests

Cinq tests vous seront distribués durant la session afin d'évaluer vos compétences sur la matière vue en classe. Chaque test vaut 5%. La date d'un test vous sera communiquée en classe la semaine précédent sa tenue. Vous n'avez pas droit à vos notes ou aux appareils électroniques lors des tests, sauf lors d'avis contraire.

Examens

L'examen intra est considéré comme un test supplémentaire. L'examen final portera principalement sur l'analyse et la conception ainsi que la matière vue en classe qui n'aura pas été couverte par les tests. À moins d'avis contraire, vous aurez droit à deux feuilles de notes recto-verso écrites à la main pour l'examen final. Aucun ordinateur, téléphone cellulaire, calculatrice ou autre appareils électroniques n'est autorisé lors des examens.

Devoirs et laboratoires

1. Les 4 premiers devoirs ont pour but de faire comprendre les principes de base de la programmation à l'aide du langage vu en cours (le C++). Le cinquième devoir vous fera pratiquer seulement l'analyse et la conception d'une solution.

Les trois devoirs suivants ont pour but de vérifier que les étudiantes et les étudiants savent analyser un problème plus complexe, concevoir une solution et finalement implémenter leur solution.

2. Les deux premiers devoirs et le cinquième doivent être effectués individuellement. Les autres travaux se feront obligatoirement en équipe de 2 personnes. Si, pour ces derniers, le travail est effectué seul ou par équipes de trois une pénalité de 50% sera imposée. Cette pénalité sera de 100% pour des équipes de plus que trois.
3. Les instructions pour la remise des devoirs seront spécifiées dans chaque énoncé.

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

4. Il y a huit (8) devoirs dans la session, mais seulement les 6 meilleurs seront comptabilisés. Le seul devoir que vous devez obligatoirement remettre est le dernier. Chaque devoir aura une pondération de 5 points pour un total de 30%.
5. Il y aura 9 laboratoires dans la session. Cinq de ces laboratoires seront choisis au hasard, corrigés et comptabilisés avec une pondération de 2 points par laboratoire pour un total de 10%.
6. La qualité du français et de la présentation est considérée dans le résultat du travail.
7. Les sujets des travaux seront disponibles sur la page WEB du cours au jour spécifié dans le plan de cours pour la réception du devoir. La remise du travail s'effectue le jour et à l'heure exigés. Le non respect de la date de remise entraîne une pénalité de 25% de la note par tranche de 8 heures de retard, à moins d'un cas exceptionnel. Il est à noter qu'un oubli ou un emploi du temps chargé n'est pas un cas exceptionnel. Il en est de même en ce qui concerne une panne électrique, d'ordinateur, d'imprimante, ou du réseau. Cela signifie qu'il faut toujours viser à terminer son travail de programmation au moins 24 heures avant la date de remise pour tenir compte des pannes possibles et de la surcharge quasi-inévitable. Ceci est un conseil qui vaut son pesant de points. Les modalités de remise de chaque travail vous seront fournies avec le sujet de chaque travail.
8. Le respect des normes départementales est impératif. Le document « Norme de documentation des programmes » [2] contient les normes du département en matière de programmation.
Cette contrainte permet de vérifier que l'étudiante ou l'étudiant sait s'astreindre à une discipline de programmation. Elle permet de plus de mieux insister sur les concepts importants du cours.
9. Vos programmes doivent pouvoir être compilés autant sous Windows que sous un système basé sur UNIX. Vous ne pouvez pas prendre pour acquis les particularités d'un système d'exploitation, votre code doit donc respecter les standards du langage. Il est quand même dans votre intérêt d'apprendre à utiliser un minimum l'environnement UNIX puisque vous aurez à l'utiliser au cours de votre bac.
De l'aide technique est disponible sur les ordinateurs des laboratoires, que ceux-ci tournent sous Windows, Linux ou Solaris. Sur le site, les étudiantes et étudiants pourront trouver des liens et de la documentation pour travailler à partir de chez eux. Cependant, aucune autre aide ne sera fournie par manque de temps et de ressources.

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 le règlement 4.2.3 s'applique à moins d'avoir obtenu personnellement l'autorisation du professeur. Cette permission peut être retirée en tout temps, si l'appareil n'est pas utilisé uniquement à des fins d'apprentissage.

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 du professeur.

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

3 Matériel nécessaire pour l'activité pédagogique

Les notes de cours sont disponibles sur le WEB.

Le manuel sur lequel est basé le cours est celui de Friedmann [4]. L'achat de ce livre n'est pas obligatoire et est laissé à votre discrétion.

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

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

Les normes de programmation du Département d'informatique sont décrites dans [2]. Vous devez absolument vous procurer ce document et le lire.

Même si l'achat du livre n'est pas nécessaire, il est par contre fortement recommandé d'avoir au moins un livre de la liste de référence.

4 Références

- [1] Yves BOUDREAU et Wacef GUERFALI : *C++ et un peu + (2e édition)*. Presses Internationales Polytechnique, 2001.
- [2] Alex BOULANGER et Félix-Antoine OUELLET : Normes de programmation pour le cours IFT159. <http://info.usherbrooke.ca/vducharme/ift159/Normes.pdf>, 2014. Aussi disponible sur le site Web du cours.
- [3] H. M. DEITEL et P. J. DEITEL : *C++ : How to program*. Prentice-Hall, 2001.
- [4] Frank L. FRIEDMAN et Elliot B. KOFFMAN : *Problem Solving, Abstraction, and Design Using C++*. Addison-Wesley, 2004.
- [5] Cay HORSTMAN et Timothy BUDD : *Big C++, Second edition*. John Wiley, 2009.

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
