

Université de
Sherbrooke

Département d'informatique
IFT 159 – Analyse et programmation
Plan d'activité pédagogique
Hiver 2022

Enseignant

Marie-Flavie Auclair-Fortier

Courriel : marie-flavie.auclair-fortier@usherbrooke.ca

Local : D4-1010-06

Téléphone : +1 819 821-8000 x62855

Disponibilités : À déterminer en début de session.

Responsable(s) : Direction du département

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

Horaire

Exposé magistral :	Lundi	8h30 à 10h20	salle D3-2039
	Jeudi	8h30 à 10h20	salle D3-2035
Exercices/laboratoires :	Mercredi	13h30 à 14h20	salle D4-1023/1017/0023/0024

Description officielle de l'activité pédagogique¹

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 classe). 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>

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 159, Analyse et Programmation, est le premier cours d'informatique des baccalauréats en science du multimédia et du jeu vidéo, 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.

Une personne ou une équipe qui se voit confier le développement d'un système doit d'abord l'analyser au complet avant de penser à programmer la solution retenue. Cette approche est privilégiée dans le cours lors des études de cas. Elle pourra aussi l'être lors des travaux pratiques.

Ce cours est aussi le premier d'une chaîne de cours d'analyse de problèmes et de programmation : en S2 IFT 339 et IFT232, en S3 IFT 320 et IFT 359, 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 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 seulement d'apprendre le C++, mais bien le processus menant à résoudre un problème en utilisant l'informatique, et ce, avec n'importe quel langage structuré.

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

1. de comprendre le fonctionnement d'un ordinateur dans le contexte de l'utilisation d'un outil d'élaboration d'une solution programmée ;
2. de lire et comprendre un énoncé de problème peu complexe ;
3. d'analyser un problème pour y offrir une solution algorithmique ;
4. d'illustrer un algorithme en utilisant la représentation appropriée ;
5. de mettre en œuvre un algorithme à l'aide d'un langage de programmation ;
6. de planifier des tests pour un programme ;
7. de vérifier le bon fonctionnement d'un programme ;
8. de respecter des normes et standards de programmation ;
9. d'utiliser les mécanismes élémentaires d'encapsulation orientée objet ;
10. de rédiger un programme en appliquant les principes de base de la dérivation par enrichissements successifs ;
11. 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	Nbr. d'heures	Objectifs	Travaux	Lectures
1	Brève introduction aux ordinateurs : Modèle pratique du calcul, modèle théorique du calcul, étapes de la mise en œuvre et de l'exécution d'un programme.	3	1		Chapitre 1 [1] Chapitre 1 [2] Section 1.1 [3]
2	Développement de programmes : Cycle de vie; phases de développement du logiciel : spécification, analyse, conception, programmation; notions de gestion d'effort (PSP0).	2	2, 3	✓	Chapitre 1 [1] Chapitre 2 [2] Chapitres 1 et 2 [3]
3	Introduction à la programmation : Éléments de base du C++, notions de programmes (variables, constantes, types, énoncés, fonctions, etc.); compilation.	2	3, 5, 7	✓	Chapitre 2 [1] Chapitre 2 [2] Chapitres 1 et 2 [3]
4	Les fonctions : Concept de modularité; analyse et conception descendante; fonctions (avec ou sans paramètre); paramètres (valeur, référence); fiabilité (validité, robustesse, assertions); utilisations avancées (expressions logiques, etc.); méthodes de tests.	8	2 à 8, 10 et 11	✓	Chapitres 3 et 6 [1] Chapitre 6 [2] Chapitre 4 [3]
5	Structures sélectives des langages : Expressions logiques; énoncé « if », énoncés composés, énoncé « if » emboîté, énoncé « switch »; notions d'analyse et de conception.	3	2 à 8, 10 et 11	✓	Chapitre 4 [1] Chapitre 4 [2] Section 2.4 [3] Chapitre 3 [3]
6	Structures itératives des langages : Concept d'itération; boucle conditionnelle et de comptage; analyse et conception; énoncés « while », « for » et « do... while »; boucles emboîtées.	3	2 à 8, 10 et 11	✓	Chapitre 5 [1] Chapitre 5 [2] Section 2.4 [3] Chapitre 3 [3]
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 à 8, 10 et 11	✓	Chapitres 7 et 9 [1] Chapitre 8 [2] Chapitres 7 et 8 [3]
8	Introduction à l'abstraction de données : Encapsulation; concepts d'abstraction de données; introduction aux classes; introduction à UML : diagramme de classes.	6	9	✓	Chapitre 10 [1] Section 11.5 [1] Chapitre 7 [2] Section 11.6 [2] Chapitre 10 [3]
9	Récursivité : Caractéristiques d'un problème récursif; exemples de problèmes récursifs.	3	3 à 5		Chapitre 12 [1] Chapitre 14 [2] Chapitre 14 [3]
10	Introduction à la complexité algorithmique : Définition; concept d'ordre de complexité; exemples.	2	3		Section 9.6 [1] Chapitre 9 [2] Section 18.3 [3]
11	Conclusion : Conclusion.	1			

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

Une semaine régulière comprendra

- quatre heures de présence en classe pour les cours magistraux et des exercices
- une heure en laboratoire pour de l'apprentissage actif, des exercices et des laboratoires

Chaque semaine, il y aura environ quatre heures d'exposés magistraux décrivant la théorie ainsi que des exemples développés au tableau ou des exercices. Il y aura aussi une heure d'exercices pratiques, d'auto-apprentissage, qui seront directement intégrés dans des séances de laboratoire et durant lesquelles des assistant.es seront présent.es et disponibles. **C'est à vous que revient la tâche de vous organiser pour vous exercer sur ce qui sera vu en cours.**

Les cours magistraux se dérouleront en partie au tableau et en partie sur diapositives électroniques.

Pour vous connecter à Moodle allez au www.usherbrooke.ca/moodle2-cours/ et suivez les instructions. Vous devez avoir un CIP.

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	Laboratoires
1	2022-01-03	1	
2	2022-01-10	2	Environnement de développement C++ et Compilation et débogage
3	2022-01-17	2 et 5	Séquences et sélections
4	2022-01-24	2 et 6	Itérations
5	2022-01-31	4	Fonctions
6	2022-02-07	4	Compilation séparée et modules
7	2022-02-14	Révision et 4	
8	2022-02-21	Examen périodique	
9	2022-02-28	Relâche	
10	2022-03-07	7	Tableaux
11	2022-03-14	8	Enregistrements
12	2022-03-21	8	Classes
13	2022-03-28	9	Récursivité
14	2022-04-04	10	
15	2022-04-11	Révision et 11	
16	2022-04-18	Examen final	

- La semaine du 10 janvier, il y aura DEUX séances de laboratoires différentes. Une à l'horaire régulier et une autre le mardi 11 janvier (heure à préciser).
- La semaine du 24 janvier, la séance de laboratoire qui devait se tenir le 26 aura plutôt lieu le 25 janvier (heure à préciser).

2.3 Évaluation

Devoirs	30 %
Laboratoires (10)	10 %
Examen intra	30 %
Examen final	30 %

POINTS BONUS : Vous aurez la possibilité d'accumuler des points bonus de participation à la communauté du cours sur les

forums Moodle, soit en posant des questions pertinentes au cours (théorie ou pratique) ou en fournissant votre aide à *l'ensemble de la communauté*. C'est la personne enseignante qui sera juge de la pertinence des questions/réponses. Ce bonus, pour un maximum de 10 %, sera calculé à la fin du cours.

2.3.1 Objectifs des

1. **Laboratoires** : mettre *rapidement* en pratique les notions vues en classe, sur des problèmes simples, à l'aide d'un langage de programmation dans un environnement supervisé au besoin.
2. **Devoirs** : mettre en pratique les notions vues en classe sur des *problèmes plus complexes*, nécessitant un certain niveau d'analyse.
 - (a) Les trois (3) 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++);
 - (b) Le quatrième devoir vous fera vous exercer seulement à l'analyse, la conception et la validation d'une solution, sans l'implémenter;
 - (c) Les deux derniers devoirs ont pour but de vous faire expérimenter des types de données plus complexes;
 - (d) Tous les devoirs vous feront expérimenter le processus de développement d'une solution informatique : analyser un problème, concevoir une solution et finalement implémenter et valider celle-ci.

2.3.2 Logistique

1. Les sujets des laboratoires et des devoirs seront disponibles sur le site Moodle. La remise de chaque travail s'effectue le jour et à l'heure exigés. Le non respect de la date de remise entraîne une pénalité de 10 % par jour de retard. 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.
2. Il y aura 10 laboratoires dans la session. Cinq (5) de ces laboratoires seront choisis au hasard, corrigés et comptabilisés avec une pondération de deux (2) points par laboratoire pour un total de 10 %.
3. Il y a six (6) devoirs dans la session. Chaque devoir aura une pondération de cinq (5) points pour un total de 30 %.
4. Les premier et quatrième devoirs doivent être effectués individuellement. Les autres travaux se feront obligatoirement en équipes de deux (2) personnes. Si, pour ces derniers, le travail est effectué seul ou en équipes de trois (3) personnes une pénalité de 50 % sera imposée. Cette pénalité sera de 100 % pour des équipes de plus de trois.
5. Si la COVID-19 le permet, le dernier devoir pourrait être sous forme d'une simulation en laboratoire, en classe et à la maison.
6. La remise des travaux se fera en général dans Moodle et sur turnin web. Les instructions pour la remise des devoirs et laboratoires seront spécifiées dans chaque énoncé.
7. À la fin de chaque devoir, un questionnaire réflexif **individuel** sera à remplir sur Moodle. Les instructions seront indiquées avec celles du devoir. Ce questionnaire doit être remplis pour chaque devoir. Il y a une partie des points du devoir qui va sur le sérieux de votre démarche réflexive.
8. Le respect des normes de programmation est impératif. Cette contrainte permet de vérifier que vous savez vous astreindre à une discipline de programmation. De plus, elle permet de mieux insister sur des concepts importants du cours. Les normes sont disponibles dans Moodle.
9. Vos programmes doivent pouvoir être compilés autant sous Windows que sous Linux. Vous ne pouvez pas prendre pour acquis les particularités d'un quelconque système d'exploitation, votre code doit donc respecter les standards du langage. Il est quand même dans votre intérêt d'apprendre un minimum à utiliser l'environnement Linux puisque vous aurez à l'utiliser au cours de votre bac.
10. De l'aide technique est disponible sur les ordinateurs des laboratoires, que ceux-ci tournent sous Windows, Linux. Sur les laboratoires, vous pourrez trouver des liens et de la documentation pour travailler à partir de votre poste personnel. Cependant, aucune autre aide ne sera fournie par manque de temps et de ressources humaines.

2.3.3 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.

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

2.3.4 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

Les dates de remise des travaux seront indiquées sur les énoncés.

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.

Note : Les forums dans Moodle devraient être privilégiés pour poser vos questions. Celles-ci pourraient être bénéfiques à l'ensemble de la classe. Voir la mention **BONUS** dans la section 2.3 - évaluations.

3 Matériel nécessaire pour l'activité pédagogique

- Les diapositives électroniques seront disponibles sur la [page Moodle de l'activité pédagogique](#) sous forme de pdf. Vous pourrez les télécharger d'avance si vous désirez les imprimer pour suivre en classe.
- Vous devez avoir une paire d'écouteurs avec prise 3.5 mm pour écouter les vidéos disponibles sur le site Moodle du cours lorsque vous êtes dans les laboratoires.

4 Références

- [1] FRIEDMAN, FRANK L. AND KOFFMAN, ELLIOT B. : *Problem Solving, Abstraction, and Design Using C++*, Sixth Edition. Addison-Wesley, 2011.
- [2] GADDIS, TONY AND WALTERS JUDY AND MUGANDA, GODFREY : *Starting out with C++ Early Objects (Ninth Edition)*. Pearson, 2017.
- [3] WALTER SAVITCH : *Problem Solving with C++ (9th Edition)*. Pearson, 2015.

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

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

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



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
