

**Département d'informatique**  
**IFT159 — Analyse et programmation**

Plan de l'activité pédagogique  
**Hiver 2018**

---

**Enseignant :** Marie-Flavie Auclair-Fortier  
Courriel : [Marie-Flavie.Auclair-Fortier@USherbrooke.ca](mailto:Marie-Flavie.Auclair-Fortier@USherbrooke.ca)  
Local : D4-1010-6  
Téléphone : (819) 821-8000 poste 62855  
Site Moodle : <https://www.usherbrooke.ca/moodle2-cours/course/view.php?id=14035>  
Disponibilité : sur rendez-vous

---

**Auxiliaire :** à venir  
Courriel :  
Disponibilité : à déterminer

---

**Professeur responsable :** Gabriel Girard

---

**Horaire :**

Exposés magistraux :	Lundi	8 h 30 à 10 h 20	salle D3-2038
	Jeudi	10 h 30 à 11 h 30	salle D3-2038
Exercices/Laboratoires :	Mardi	13 h 30 à 14 h 30	salles D4-1017 et D4-1023.

---

**Description officielle de l'activité pédagogique<sup>1</sup>**

**Cible(s) 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'exposés magistraux par semaine  
1 heure d'exercices par semaine  
5 heures de travail personnel par semaine

**Particularités :** aucune

---

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

## 1. Présentation

Cette section présente les objectifs 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. Objectifs 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<sup>2 3 4</sup>

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		
<b>Total</b>		<b>48</b>		

<sup>2</sup> Les heures associées à un thème particulier incluent les heures d'exercices pour un total de 48 heures (12 semaines de quatre heures).

<sup>3</sup> 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.

<sup>4</sup> 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 quatre heures de présence en classe pour les cours magistraux. Une heure supplémentaire d'assistance en laboratoire est prévue à l'horaire.

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. 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.

Les cours magistraux se dérouleront en partie au tableau et en partie sur acétates électroniques. Les transparents électroniques seront disponibles sur la [page Moodle de l'activité pédagogique](#).

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

### 2.2. Calendrier

	Semaine du	Thème	Laboratoire	Lectures dans <a href="#">ref. 4</a>
1	08 janvier	Intro		Chap. 1-2
2	15 janvier	1-2	Présentation de l'environnement	Chap. 2-3
3	22 janvier	2-3	Environnement et compilation	Chap. 3
4	29 janvier	3-4	Séquences et fonctions	Chap. 4
5	05 février	4-5	Sélection/itération	Chap. 4-5
6	12 février	5	Tableaux et vecteurs	Chap. 5
7	19 février	5-6-7	Classes	Chap. 5-6 et sect. 9.1, 9.2, 9.4, 9.5 et 11.1
8	26 février	Examen périodique		
9	5 mars	Semaine de lecture		
10	12 mars	6	Tableau	Chap. 6
11	19 mars	7	Récurtivité	Sect. 9.3, 9.7 et 9.8
12	26 mars	7-8	Robustesse	Chap. 10 et sect. 9.3, 9.7 et 9.8
13	2 avril	8-9	Librairies	Chap. 10 et sect. 9.3, 9.7 et 9.8
14	9 avril	9-10-11		Chap. 12 et sect. 9.6 et 11.7
15	16 avril	9-10-11		Chap. 12 et sect. 9.6 et 11.7
	du mardi 17 au vendredi 27 avril	Examen final		

## 2.3. Évaluation

Laboratoires :	10 %
Devoirs :	30 %
Examen périodique:	25 %
Examen final:	35 %

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages 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.

## 2.4. Travaux

### Échéancier

Devoir	Thème	Réception	Remise
1	Premier programme	22 janv. 2018	29 janv. 2018
2	Fonctions	29 janv. 2018	5 févr. 2018
3	Sélection et itération	5 févr. 2018	12 févr. 2018
4	Tableaux et vecteurs	12 févr. 2018	19 févr. 2018
5	Analyse/Conception	19 févr. 2018	26 févr. 2018
6	Classes	26 févr. 2018	12 mars 2018
7	Récursivité	12 mars 2018	26 mars 2018
8	Programme complexe	26 mars 2018	16 avr. 2018

### Devoirs et laboratoires

1. Les quatre (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 vous savez analyser un problème plus complexe, concevoir une solution et finalement implémenter celle-ci.
2. 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 sur ce même site. 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. Ceci est un conseil qui vaut son pesant de points. Des instructions particulières seront données pour chacun des travaux. La remise des travaux se fera en général dans Moodle.
3. Les instructions pour la remise des devoirs et laboratoires seront spécifiées dans chaque énoncé.
4. Il y a huit (8) devoirs dans la session, mais seulement les six (6) meilleurs seront comptabilisés. Le seul devoir que vous devez obligatoirement remettre est le dernier. Chaque devoir aura une pondération de cinq (5) points pour un total de 30%.
5. Il y aura 10 laboratoires dans la session. Cinq de ces laboratoires seront choisis au hasard, corrigés et comptabilisés avec une pondération de deux (2) points par laboratoire pour un total de 10%.
6. Les premier, deuxième et cinquième 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.

7. La qualité du français et de la présentation générale est considérée dans le résultat du travail.
8. Le respect des normes de programmation 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 vous savez vous astreindre à une discipline de programmation. De plus, elle permet 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 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 UNIX 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 ou Solaris. Sur les laboratoires, vous pourrez trouver des liens et de la documentation pour travailler à partir de chez vous. Cependant, aucune autre aide ne sera fournie par manque de temps et de ressources humaines.

## Qualité du français et de la présentation

Jusqu'à 5 % du total des points alloués pour un travail donné est rattaché à la qualité du français. De plus, Conformément aux articles 36, 37 et 38 du [règlement de la Faculté des sciences sur l'évaluation des apprentissages](#), l'enseignant peut retourner à l'étudiante ou l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

### 2.5. Remarques sur le plagiat

Un document dont le texte et la structure se rapporte à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat, défini dans le Règlement des études comme « le fait, dans une activité pédagogique évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui. ». Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du [Règlement des études](#) : « tout acte ou manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique. » À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera référé 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 reste le fait de votre équipe. De même, si l'utilisation de morceau de code source ainsi que de documentation provenant du net est autorisée, il ne faut pas que le devoir copie simplement le contenu complet ou partiel du document de référence, mais que l'étudiant ou l'étudiante démontre la compréhension de la partie utilisée. L'enseignant peut, en cas de doute, 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.

### 2.6. Utilisation d'appareils électroniques et du courriel

Dans cette activité pédagogique, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite 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 4.2.3 (ref. 8), toute utilisation d'appareils de captation de la voix ou de l'image exige la permission du professeur.

Note : L'utilisation du courrier électronique et des forums Moodle est recommandée pour poser vos questions.

## 3. Matériel nécessaire pour le cours

Les transparents électroniques à télécharger au fur et à mesure sur le site Moodle du cours.

Les notes manuscrites qui sont de votre responsabilité.

Les normes de programmation du Département d'informatique. Vous devez **absolument** vous procurer ce document et le lire :

1. Alex BOULANGER et Félix-Antoine OUELLET  
Normes de programmation pour le cours IFT159. [Disponible sur le site Moodle du cours.](#)

#### 4. Bibliographie

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

Ouvrages de référence que l'on peut notamment consulter à la bibliothèque.

2. Yves BOUDREAU et Wacef GUERFALI  
C++ et un peu + (2e édition). Presses Internationales Polytechnique, 2001.
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.

##### 4.1. Liens utiles

6. Bibliothèque de l'Université de Sherbrooke  
[www.usherbrooke.ca/biblio](http://www.usherbrooke.ca/biblio)
7. Faculté des sciences, Université de Sherbrooke  
Politique d'évaluation des apprentissages  
[www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-devaluation/](http://www.usherbrooke.ca/sciences/intranet/informations-academiques/reglement-devaluation/)
8. Université de Sherbrooke  
Règlement des études  
[www.usherbrooke.ca/accueil/fr/direction/documents-officiels/reglements/](http://www.usherbrooke.ca/accueil/fr/direction/documents-officiels/reglements/)
9. Autres liens sur le [site Moodle](#) de l'activité pédagogique



---

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

---

### **Extrait du Règlement des études**

#### **8.1.2 Relativement aux activités pédagogiques**

L'expression délit désigne d'abord tout acte ou toute manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique.

Sans restreindre la portée générale de ce qui précède, est considéré comme un délit :

- a) la substitution de personnes ou l'usurpation d'identité lors d'une activité évaluée ou obligatoire;
- b) le plagiat, soit le fait, dans une activité évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui;
- c) l'obtention par vol ou par toute autre manœuvre frauduleuse de document ou de matériel, la possession ou l'utilisation de tout matériel non autorisé avant ou pendant un examen ou un travail faisant l'objet d'une évaluation;
- d) le fait de fournir ou d'obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour un examen ou un travail faisant l'objet d'une évaluation;
- e) le fait de soumettre, sans autorisation préalable, une même production comme travail à une deuxième activité pédagogique;
- f) la falsification d'un document aux fins d'obtenir une évaluation supérieure dans une activité ou pour l'admission à un programme.

#### **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 (autoplégat)

---

## **Autrement dit : mentionnez vos sources.**

---