

**Département d'informatique****IFT 232 – Méthodes de conception orientées objet****Plan d'activité pédagogique
Hiver 2019**

Enseignant**Mikaël Fortin**

Courriel :	mikael.fortin@usherbrooke.ca
Local :	D4-2004
Téléphone :	(819) 821-8000 poste 66106
Site :	
Disponibilité :	À déterminer

Responsable(s) : Sylvain Giroux

Horaire

Exposé magistral :	Lundi	8h30 à 10h20	salle D3-2038
	Mercredi	8h30 à 10h20	salle D3-2038

Les séances de laboratoire auront lieu une semaine sur deux, durant une des deux périodes de cours, au D4-1017 ou D4-0023, selon les disponibilités.

Description officielle de l'activité pédagogique¹

Cibles de formation : Spécifier, concevoir et tester des composants logiciels. Tester l'intégration des composants. Mesurer la qualité de la conception. Appliquer le processus de conception au sein d'un cadre documenté et normalisé.

Contenu : Types abstraits algébriques. Critères de conception. Encapsulation, héritage et polymorphisme. Critères de composition en classes, schémas de conception (design patterns) et cadres d'application (frameworks). Documentation de la conception avec la notation UML. Techniques de revue de conception. Tests unitaires. Programmation par composants. Mesure de qualité de la conception. Introduction à la programmation agile.

Crédits 3
Organisation 3 heures d'exposé magistral par semaine
1 heure d'exercices par semaine
5 heures de travail personnel par semaine
Préalable GIF600 ou IFT159
Particularités Aucune

1. <https://www.usherbrooke.ca/admission/fiches-cours/ift232.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

La programmation orientée objet (POO) est un paradigme informatique considéré comme une des innovations les plus importantes en génie logiciel². À travers SIMULA, les origines de la POO remontent aussi loin qu'en 1967. La POO a donné par la suite naissance à d'autres innovations significatives en informatique comme le modèle MVC (1980), les schémas de conception (1991) et la refactorisation (1993). La POO repose sur cinq concepts fortement intégrés qu'elle articule en un tout cohérent : objet (encapsulation), classe (instanciation et réutilisation), message (communication), héritage (réutilisation) et méthode (comportement). Comme tout paradigme, la POO a développé sa propre culture, sa propre façon de concevoir le monde : on n'écrit pas un programme objet comme on écrit un programme fonctionnel.

1.2 Cibles de formation spécifiques

L'objectif du cours IFT232 est de développer un point de vue intégré et vertical de la culture de la conception par objets : de la conception jusqu'au programme final, à sa maintenance et à son évolution. A la fin du cours, un étudiant ou une étudiante sera capable d'analyser en termes de POO un problème de taille moyenne, de concevoir une solution adaptée, de la programmer en objets et de la faire évoluer. Il ou elle devra être en outre capable de la valider et d'en juger la qualité à l'aide de critères bien définis. 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 les concepts fondamentaux de la conception et de la programmation par objets.

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

1. Maîtriser un langage de spécification ;
2. Maîtriser les concepts de base de la programmation par objets ;
3. Maîtriser le processus de refactorisation d'un programme ;
4. Maîtriser la notion de schémas de conception ;
5. Maîtriser les processus de tests ;
6. Connaître diverses approches de la conception orientée objet ;
7. Être capable d'évaluer la qualité de conception d'un programme.

1.3 Contenu détaillé

Thème	Contenu	Heures	Cibles de formation	Travaux
1	Programmation par objets : Les principes de base de la programmation par objets : instanciation, encapsulation, héritage, polymorphisme, transmission de messages, ...; Application aux langages de programmation	8	2	√

2. David A. Wheeler, « The Most Important Software Innovations », révision du 22 juillet 2011, <http://www.dwheeler.com/innovation/innovation.html>

2	Spécifications : Types abstraits algébriques; Programmation par contrats; Spécification de programmes: préconditions, postconditions, invariant, ...	3	5, 6	
3	Tests : Test unitaires; Application aux langages de programmation	4	5	✓
4	Documentation de la conception (UML)	4	1	
5	Refactorisation : Critères de conception et qualité de la conception; Techniques de refactorisation	4	3, 4, 7	✓
6	Programmation par composants : Exemple d'application.	4	4, 6	
7	Schémas de conception, principes généraux	2	4, 7	✓
8	Schémas de conception : State	4	4, 7	✓
9	Schémas de conception : Observer	4	4, 7	✓
10	Schémas de conception: Factory Method et Template Method	4	4, 7	✓
11	Schémas de conception: Command	4	4, 7	✓
12	Cadre de travail : Principes et applications; Étude de cas	3	4, 6	

1. Les heures associées à un thème particulier incluent les heures d'exercices pour un total de 48 heures (12 semaines de quatre heures).
2. Le cours doit comprendre au moins quatre travaux pratiques couvrant tous les sujets marqués ✓ dans le tableau.

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 4 heures de présence en classe. Sur toute la session, il y aura en moyenne trois heures de cours dit théoriques et une heure d'exercices. Des séances de laboratoire de 2 heures alterneront avec les blocs de cours théoriques. Le cours a une approche pratique et met l'accent sur l'implémentation des solutions conçues. Tout le matériel pédagogique sera placé sur le lecteur réseau public.

2.2 Calendrier

Semaine	Date	Thèmes	Travaux	Laboratoires
1	07/01/19	1	Début Jarvis partie 1	Labo 0 : Introduction à Java
2	14/01/19	1		Labo 1 : Programmation orientée objet
3	21/01/19	2,3		
4	28/01/19	3	Remise Jarvis Partie 1, Début Jarvis Partie 2	
5	04/02/19	3,5		Labo 2 : Tests unitaires
6	11/02/19	5		Labo 3 : Réusinage partie 1
7	18/02/19	5,8		Labo 4: Réusinage partie 2
8	25/02/19	Examen périodique	Remise Jarvis partie 2	
9	04/03/19	Relâche		
10	11/03/19	7	Début du travail de synthèse, partie 1	
11	18/03/19	9		Labo 5 : Patron Observer
12	25/03/19	10	Remise travail de synthèse, partie 1, début partie 2	Labo 6 : Templates et Factories
13	01/04/19	11		Labo 7 : Patron Command
14	08/04/19	11,12	Remise travail de synthèse, partie 2	Labo X: Révision
15	15/04/19	Finaux		

2.3 Évaluation

Jarvis, parties 1 et 2 :	8 %
Laboratoires (7) :	20 %
Travail d'intégration, parties 1 et 2 :	12 %
Examen périodique :	25 %
Examen final :	35 %

Conformément au règlement facultaire d'évaluation des apprentissages³ 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.

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*⁴ 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 plagia, l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'il considère comme étant plagié. En

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

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

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

Les laboratoires sont toujours à remettre le lundi suivant la séance de laboratoire. Voir le calendrier pour les dates de début et de fin de chaque devoir.

Directives particulières :

1. Travaux pratiques

Les travaux pratiques sont effectués seul(e) ou par équipe de deux étudiant(e)s. Les travaux pratiques sont réalisés en Java. L'environnement de programmation intégré Eclipse (<http://www.eclipse.org/>) est disponible au département d'informatique. L'environnement de tests utilisé est JUnit (<http://www.junit.org/>). Les travaux pratiques peuvent être réalisés sous tout autre environnement, mais le code et les tests devront fonctionner sous Eclipse et Java SE 7 et JUnit 4 au moment de la remise du travail. Les énoncés des travaux seront rendus disponibles sur le lecteur réseau public. La remise du travail (code, tests et autres documents) sera effectuée par turnin web. La remise du travail s'effectue au jour et à l'heure exigés.

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁵, 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 autorisé. Cette permission peut être retirée en tout temps si leur usage entraîne des abus.

Dans le règlement universitaire des études, section 4.2.3⁶, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission du professeur.

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

3 Matériel nécessaire pour l'activité pédagogique

Les manuels et articles en référence ne sont pas obligatoires. Toutefois les manuels recommandés sont [7], [3], [8] et [13]. Le langage utilisé dans le cours est Java. Les projets peuvent être réalisés dans le langage à objets le plus approprié à la nature du projet, e.g. existence d'une bibliothèque donnée.

4 Références

[1] Extreme programming, a gentle introduction. <http://www.extremeprogramming.org/>.

[2] Refactoring to patterns catalog. <http://www.industriallogic.com/xp/refactoring/catalog.html>.

[3] L. AUDIBERT : *UML 2 : de l'apprentissage à la pratique*. Ellipses, Paris, 2009.

[4] K. BECK : *Extreme programming explained*. Addison-Wesley, Reading, MA, 2000.

[5] K. BECK : *Test driven development : by example*. Addison-Wesley, Boston, 2003.

[6] H. BERSINI : *L'orienté objet*. Eyrolles, Paris, 2^e édition, 2004.

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

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

- [7] J. BLOCH : *Effective Java Programming Language Guide*. Addison-Wesley, 2^e édition, 2008.
- [8] J.-L. BÉNARD : *Gestion de projet eXtreme Programming*. Eyrolles, 2005.
- [9] T. BUDD : *An introduction to object-oriented programming*. Addison-Wesley, 3^e édition, 2002.
- [10] B. CHARROUX, A. OSMANI et Y. THIERRY-MIEG : *UML 2 : pratique de la modélisation*. Addison-Wesley, 3^e édition, 2010.
- [11] M. FOWLER : *Refactoring, improving the design of existing code*. Addison-Wesley, Reading, MA, 2000.
- [12] E. FREEMAN, E. FREEMAN et K. et B. BATES : *Design patterns : tête la première*. O'Reilly, Paris, 2005.
- [13] E. GAMMA, R. HELM, R. JOHNSON et J. VLISSIDES : *Design Patterns*. Addison-Wesley, 1994.
- [14] J. KERIEVSKY : *Refactoring to patterns*. Addison-Wesley, Boston, 2005.
- [15] C. LARMAN : *UML 2 et les design patterns*. Pearson Education, 3^e édition, 2005.
- [16] S. R. SCHACH : *Object-oriented and classical software engineering*. McGraw-Hill, 8^e édition, 2011.



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
