



Université de
Sherbrooke

Département d'informatique

IFT 232 – Méthodes de conception orientée objet

Plan d'activité pédagogique

Hiver 2024

Enseignants

	Mikaël Fortin	Hanan Bentaleb
Courriel :	Mikael.Fortin@USherbrooke.ca	Hanan.Bentaleb@USherbrooke.ca
Local :	D4-2004	
Téléphone :	+1 819 821-8000 x66106	
Disponibilités :	Sur rendez-vous	

Responsable(s) : Sylvain Giroux

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

Horaire

Groupe 1 :	Exposé magistral :	Mercredi	8h30 à 10h20	salle D3-2041
	Exercices/laboratoires :	Jeudi	10h30 à 12h20	salle D4-1023/D4-1017
Groupe 2 :	Exposé magistral :	Mercredi	13h30 à 15h20	salle D3-2041
	Exercices/laboratoires :	Jeudi	13h30 à 15h20	salle D4-1023/D4-1017
Groupe 18 :	Exposé magistral :	Mardi	10h00 à 11h50	salle L1-6680/L1-5650/L1-5670/L1-6670
	Exercices/laboratoires :	Mercredi	10h00 à 11h50	salle L1-6670

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	IFT 159
Particularités	Aucune

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

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

La programmation orientée objet (POO) est un paradigme informatique considéré comme une des innovations les plus importantes en génie logiciel [3]. 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. À 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 :

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

1.3 Contenu détaillé

Thème	Contenu	Nbr. d'heures	Objectifs	Travaux	Lectures
1	Programmation par objets : Les principes de base de la programmation par objets : instanciation, encapsulation, héritage, polymorphisme, transmission de messages, etc. ; Application aux langages de programmation.	8	2	✓	[2] [8] chapitre 7
2	Spécifications : Types abstraits algébriques ; Programmation par contrats ; Spécification de programmes : préconditions, postconditions, invariant, etc.	3	5 et 6		[8] chapitre 7
3	Tests : Tests unitaires ; Application aux langages de programmation.	4	5	✓	[5]
4	Documentation de la conception : Documentation de la conception (UML).	4	1		[1]
5	Refactorisation : Critères de conception et qualité de la conception ; Techniques de refactorisation.	4	3, 4 et 7	✓	[6]
6	Programmation par composants : Exemple d'application.	4	4 et 6		[9] chapitre 3
7	Schémas de conception : Principes généraux.	2	4 et 7	✓	[4] chapitre 1
8	Schémas de conception - État	4	4 et 7	✓	[4] chapitre 5
9	Schémas de conception - Observateur	4	4 et 7	✓	[4] chapitre 5
10	Schémas de conception - Fabrique et Patron de méthode	4	4 et 7	✓	[4] chapitres 3 et 5
11	Schémas de conception - Commande	4	4 et 7	✓	[4] chapitre 5
12	Cadre de travail : Principes et applications ; Étude de cas.	3	4 et 6		[7]

1. Le cours doit comprendre au moins quatre 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 comprend quatre 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 deux heures alternent 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	Thème
1	2024-01-08	1
2	2024-01-15	1
3	2024-01-22	1 et 4
4	2024-01-29	1
5	2024-02-05	1 et 2
6	2024-02-12	3
7	2024-02-19	5
8	2024-02-26	Examen périodique
9	2024-03-04	Relâche
10	2024-03-11	5, 8 et 12
11	2024-03-18	6 et 7
12	2024-03-25	9
13	2024-04-01	10
14	2024-04-08	11
15	2024-04-15	Examen final
16	2024-04-22	Examen final

2.3 Évaluation

Devoirs (2)	10 %
Laboratoires (7)	30 %
Examen intra	15 %
Examen intra	15 %
Examen final	30 %

Les laboratoires sont conçus pour être réalisés partiellement durant la séance de laboratoire. Ils comprennent un jeu d'exercices plus basiques réalisables rapidement, ainsi que des exercices plus avancés qui peuvent être terminés durant la semaine suivant la séance de laboratoire. L'évaluation concerne principalement la forme finale du code en termes d'organisation, ainsi que son fonctionnement vérifié par des tests.

Les devoirs sont des travaux plus longs (1 à 3 semaines) où l'évaluation comprend le fonctionnement du code soumis avec des tests, mais également, les choix d'implémentation.

Les exigences précises sont présentées sur les énoncés des travaux et laboratoires.

Des remises en retard sont admissibles, moyennant une pénalité.

Les examens visent à évaluer la compréhension des notions théoriques, ainsi que la capacité à comprendre, modifier et produire du code rapidement.

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

Les dates de remise des travaux seront indiquées sur turnin web.

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 : Je réponds aux questions posées par courriel à l'extérieur des périodes de cours.

3 Matériel nécessaire pour l'activité pédagogique

Les notes de cours seront disponibles sur le lecteur public.

4 Références

- [1] B. CHARROUX AND A. OSMANI AND Y. THIERRY-MIEG : *UML 2 : pratique de la modélisation*. Addison-Wesley, 3^e édition, 2010.
- [2] COINTE, PIERRE : Metaclasses are first class : the ObjVlisp model, 1987.
- [3] DAVID A. WHEELER : The Most Important Software Innovations. <http://www.dwheeler.com/innovation/innovation.html>, révisé le 2019-03-16.
- [4] E. GAMMA AND R. HELM AND R. JOHNSON AND J. VLISSIDES : *Design Patterns*. Addison-Wesley, 1994.

²https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Etudiants_actuels/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/Etudiants_actuels/Etudiants_actuels/Informations_academiques_et_reglements/Sciences_Reglement_complementaire.pdf

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

- [5] K. BECK : *Test driven development : by example*. Addison-Wesley, Boston, 2003.
- [6] M. FOWLER : *Refactoring, improving the design of existing code*. Addison-Wesley, Reading, MA, 2000.
- [7] ROBERTS, DON AND JOHNSON, RALPH AND OTHERS : Evolving frameworks : A pattern language for developing object-oriented frameworks. *Pattern languages of program design*, 3:471–486, 1996.
- [8] S. R. SCHACH : *Object-oriented and classical software engineering*. McGraw-Hill, 8^e édition, 2011.
- [9] WANG, ANDY JU AN AND QIAN, KAI AND OTHERS : *Component-oriented programming*, volume 319. Wiley Online Library, 2005.

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
