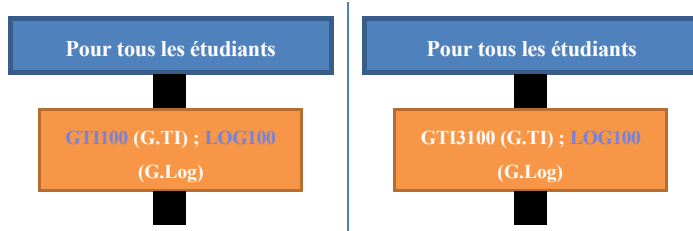


## PLAN DE COURS

Hiver 2015

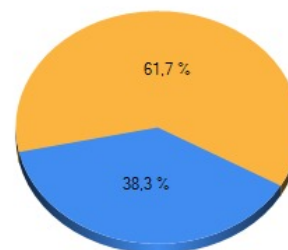
LOG121 : Conception orientée objet (4 crédits)

### Préalables



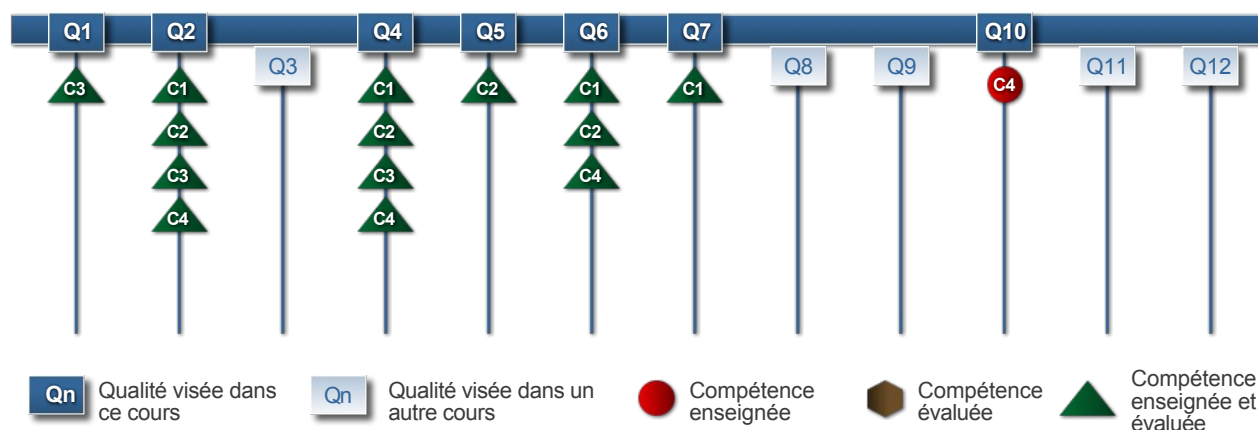
### Unités d'agrément

Total d'unités d'agrément : 64,8



Science du génie : 24,8  
Conception Ingénierie : 40

### Qualités de l'ingénieur



## Descriptif du cours

Au terme de ce cours, l'étudiant sera en mesure :

- de concevoir des systèmes orientés objet en appliquant des principes et heuristiques de base d'une bonne conception;
- d'analyser des problèmes de conception;
- de choisir et d'appliquer des patrons de conception.

Conception orientée objet : identification des objets, modélisation des objets et de leur interaction, notions de cohésion d'un objet et de couplage entre objets. Patrons de conception : compréhension des problèmes liés à la conception, choix et application des patrons appropriés résolvant ces problèmes et compréhension des conséquences d'utilisation de ces patrons. Comparaison de différents choix de conception : ajouter un comportement par composition versus l'ajouter par héritage. Notation UML : diagramme de classes, diagramme d'objets et diagramme de séquences. Notions de test unitaire et de cadre de développement.

Séances de laboratoire portant sur la conception et la mise en œuvre des projets en appliquant les principes et patrons de conception vus en classe. Utilisation de la notation UML pour documenter la conception.

## Objectifs du cours

À la fin de ce cours, l'étudiant devrait être capable de :

- distinguer entre les disciplines de conception et de programmation;
- concevoir, programmer, dériver et tester des logiciels en langage Java;
- utiliser le langage UML pour décrire la conception d'un système;
- comprendre et apprécier les notions de couplage et de cohésion et leur impact sur la conception de logiciels;
- identifier et programmer des patrons de conception;
- connaître et reconnaître les problèmes que les patrons de conception sont censés résoudre.

## Stratégies pédagogiques

Il y aura trois (3) heures de cours magistral par semaine, pendant treize (13) semaines, portant sur les principes de conception de logiciels. Ces principes seront accompagnés d'exemples concrets, d'exercices pratiques et/ou de quiz.

Il y a trois (3) heures de laboratoire par semaine qui permettront à l'étudiant d'appliquer les concepts théoriques vus en classe.

## Utilisation d'appareils électroniques

Aucun appareil utilisé

## Coordonnées de l'enseignant

| Groupe | Nom                | Activité          | Courriel   | Local  | Disponibilité |
|--------|--------------------|-------------------|--|--------|---------------|
| 01     | Dominic St-Jacques | Activité de cours | <a href="mailto:cc-Dominic.St-Jacques@etsmtl.ca">cc-Dominic.St-Jacques@etsmtl.ca</a> |        |               |
| 02     | Francis Cardinal   | Activité de cours | <a href="mailto:cc-Francis.Cardinal@etsmtl.ca">cc-Francis.Cardinal@etsmtl.ca</a>     | A-3466 |               |
| 03     | Francis Cardinal   | Activité de cours | <a href="mailto:cc-Francis.Cardinal@etsmtl.ca">cc-Francis.Cardinal@etsmtl.ca</a>     | A-3466 |               |

## Cours

- Concept de la programmation par objet et Java (3 heures)
- UML (10 heures)
- Test unitaire et cadriciel : JUnit, Collections, Swing (3 heures)
- Interface à l'utilisateur et règles de conception (3 heures)
- Modèle/vue/contrôleur et les patrons de conception (20 heures)

\* La matière ne sera pas nécessairement présentée dans cet ordre. Toutefois, l'ensemble des sujets sera présenté dans le cadre de ce cours.

## Laboratoires et travaux pratiques

Les laboratoires, sauf le premier, sont réalisés en équipe de plusieurs étudiant(e)s. Il y a quatre (4) travaux prévus pour les laboratoires :

- Laboratoire 1 (Individuel)
  - Concevoir et implémenter une application en Java : interface graphique (swing), fil d'exécution (« thread »), processus communiquant (« socket »), expression régulière (« regex »). Appliquer des règles de conception. Documenter des décisions de conception à l'aide d'UML. Utiliser l'environnement de développement intégré Eclipse. (9 heures)
- Laboratoire 2
  - Réutiliser le code source d'une application en Java. Concevoir et implémenter des nouvelles fonctionnalités : structures de données, algorithme de tri. (6 heures)
- Laboratoire 3
  - Concevoir, implémenter et tester un cadriciel en Java. Appliquer les patrons « méthode template » et « stratégie ». Utiliser le cadriciel JUnit. Développer des tests unitaires. (9 heures)
- Laboratoire 4
  - Concevoir et implémenter selon le modèle MVC (« modèle-vue-contrôleur ») une application en Java en appliquant les patrons « commande », « memento », « observateur », et d'autres. (12 heures)

## Utilisation d'outils d'ingénierie

L'étudiant se familiarise avec des environnements de développement tels que Eclipse, des compilateurs/interpréteurs pour le langage Java, des débogueurs, des outils de gestion de logiciels, et des outils de modélisation supportant le langage UML.

## Évaluation

|                            |      |
|----------------------------|------|
| Laboratoires               | 40 % |
| Quiz / devoirs / exercices | 15 % |
| Examen intra:              | 20 % |

- Groupes 1 et 3 : **mercredi 18 février 2015**
- Groupe 2 : **vendredi 20 février 2015**

|              |      |
|--------------|------|
| Examen final | 25 % |
|--------------|------|

Pour réussir ce cours, l'étudiant(e) doit avoir obtenu au moins 50% de moyenne dans les travaux individuels.

## Politique de retard des travaux

Tout travail remis en retard se verra automatiquement attribuer la note zéro.

## Absence à un examen

- Pour les départements à l'exception du SEG :

Dans les cinq (5) jours ouvrables suivant la tenue de son examen, l'étudiant devra justifier son absence d'un examen durant le trimestre auprès de la coordonnatrice – Affaires départementales qui en référera au directeur du département. Pour un examen final, l'étudiant devra justifier son absence auprès du Bureau du registraire. Toute absence non justifiée par un motif majeur (maladie certifiée par un billet de médecin, décès d'un parent immédiat ou autre) à un examen entraînera l'attribution de la note zéro (0).

- Pour SEG :

Dans les cinq (5) jours ouvrables suivant la tenue de son examen, l'étudiant devra justifier son absence auprès de son enseignant. Pour un examen final, l'étudiant devra justifier son absence auprès du Bureau du registraire. Toute absence non justifiée par un motif majeur (maladie certifiée par un billet de médecin, décès d'un parent immédiat ou autre) à un examen entraînera l'attribution de la note zéro (0).

## Plagiat et fraude

Les clauses du « Chapitre 10 : Plagiat et fraude » du « Règlement des études de 1er cycle » s'appliquent dans ce cours ainsi que dans tous les cours du département. Afin de se sensibiliser au respect de la propriété intellectuelle, tous les étudiants doivent consulter la page Citer, pas plagier !

<http://www.etsmtl.ca/Etudiants-actuels/Baccalaureat/Guichet-interactif/Citer-pas-plagier>

## Documentation obligatoire

HORSTMANN, Cay, *Object-Oriented Design and Patterns*. Second Edition, Wiley, 2006.

## Ouvrages de références

### Optionnelles

HORTON, I., *Beginning Java 2. SDK 1.5 Edition*, Wrox Press Ltd., 2005.

GRAND, M., *Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML*. Vol. 1, 2<sup>nd</sup> Edition, New York, Wiley, 2002.

### Complémentaires

GAMMA, E., HELM, R., JOHNSON, R. & VLISSIDES, J., *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995 (aussi disponible en français).

SHALLOWAY, A. & TROTT, J., *Design Patterns Explained: A New Perspective on Object-Oriented Design*. Addison-Wesley, 2002.

CAMPIONE, M. & WALRATH, K., *The Java Tutorial*. Second Edition, Addison-Wesley, 1998.

ECKEL, B., *Thinking in Java*. Upper Saddle River (NJ), Prentice Hall, 2000

<http://www.mindview.net/Books/TIJ/>.

RUMBAUGH, J., JACOBSON, I., & BOOCH, G., *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999.

## Adresse internet du site de cours et autres liens utiles

<https://cours.etsmtl.ca/log121/>