

PLAN DE COURS

Automne 2020

INF111 : Programmation orientée-objet (hors programme) (4 crédits)

Modalités de la session d'automne 2020

Pour assurer la tenue de la session d'automne 2020, les modalités suivantes seront appliquées :

La plupart des cours de la session d'automne seront donnés à distance. Les autres seront donnés en présence. Cette information vous a déjà été communiquée.

L'étudiant inscrit à un cours à distance doit avoir accès à un ordinateur, un micro, une caméra et un accès à internet, idéalement de 10Mb/s ou plus.

Les cours à distance pourraient être enregistrés, à la discrétion de l'ÉTS. Le cas échéant, les enregistrements de cours pourraient notamment être rendus accessibles aux étudiants par le biais notamment du portail de l'ÉTS.

La notation des cours sera la notation régulière prévue aux règlements des études de l'ÉTS.

Pour les cours à distance, les examens (intra, finaux) se feront normalement à distance. Leur surveillance se fera à l'aide de la caméra et du micro de l'ordinateur et pourrait être enregistrée. Ceci est nécessaire pour se conformer aux exigences du Bureau canadien d'agrément des programmes de génie (BCAPG) afin d'assurer la validité des évaluations.

Le contexte actuel oblige bien sûr l'ÉTS à envisager la possibilité d'une deuxième vague de la pandémie de COVID-19, laquelle pourrait entraîner, après le début de la session d'automne 2020, un resserrement des directives et recommandations gouvernementales. Nous vous assurons que l'ÉTS se conformera aux règles en vigueur afin de préserver la santé publique et que, si requis, elle pourrait aller jusqu'à interdire l'accès physique au campus universitaire et ordonner la dispense en ligne de toutes les activités d'enseignement et d'évaluation pour la durée restante de la session d'automne 2020.

Des exigences additionnelles pourraient être spécifiées par l'ÉTS ou votre département, suivant les particularités propres à votre programme.

Si vous ne consentez pas aux modalités décrites précédemment, vous devez vous désinscrire de vos cours avant le 13 septembre et vous pourrez être remboursé.

Pour les nouveaux étudiants inscrits au programme de baccalauréat uniquement, vous devez vous désinscrire avant le 25 septembre et vous pourrez être remboursé.

En demeurant inscrit, vous acceptez les modalités particulières de la session d'automne 2020.

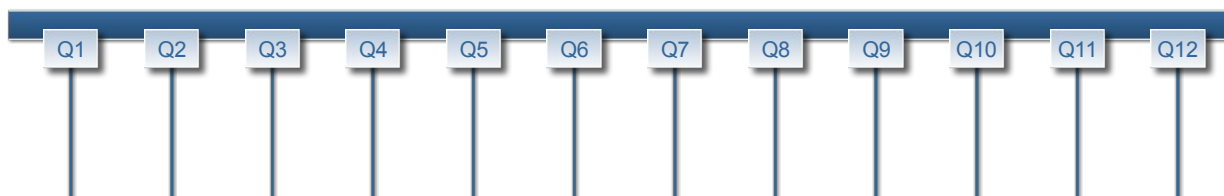
Préalables

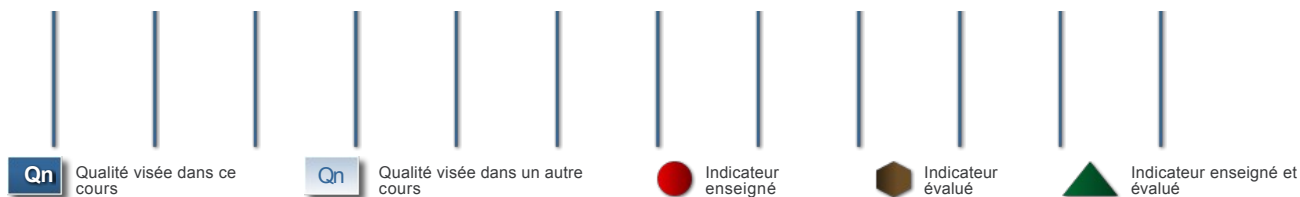
Aucun préalable requis

Unités d'agrément

Données non disponibles

Qualités de l'ingénieur





Descriptif du cours

Cours destiné aux étudiants ayant déjà suivi un cours de programmation. Il permet d'apprendre et de pratiquer les principes de base de la programmation orientée objet (encapsulation, héritage, composition et polymorphisme). Le langage de programmation utilisé est le même que pour les cours de conception suivants.

À la suite d'une présentation de base du langage utilisé et d'algorithmes de tri (sélection, insertion et bulle) et de fouille binaire, l'étudiant acquiert des principes de programmation avancée comme l'implémentation des types de données abstraits de base telles qu'une pile, une file et une liste (avec et sans position courante), autant avec tableau statique qu'avec chaînage dynamique (simple et double). Il acquiert également des notions orientées objet à l'aide de la gestion et la levée d'exception, l'utilisation de collections de base offertes par le langage utilisé (exemple : Vector, ArrayList et linkedList de Java), l'écriture de classe interne et leur avantage, l'utilisation de composants graphiques pour la construction d'interfaces utilisateurs telles que bouton, étiquette (label), panneau (panel), cadre (frame) en plus de la gestion d'événements par écouteur (listener). Le tout avec de bonnes pratiques de programmation utilisées et reconnues.

Séances de laboratoire permettant l'application des notions de programmation.

Objectifs du cours

Ce cours a pour objectif d'enseigner les principes de la programmation orientée-objet (encapsulation, héritage et polymorphisme) en insistant sur des notions de programmation plus approfondies que les types primitifs (int, char, ...), les structures de contrôle et les structures de données de base tels que les tableaux.

De façon plus spécifique, ce cours devra permettre à l'étudiant de :

- Utiliser un environnement de programmation Java (BlueJ, Eclipse, JCreator ou autres).
- Apprendre à programmer dans le paradigme orienté-objet avec le langage Java.
 - Comprendre les relations d'héritage, d'agrégation et de composition.
 - Comprendre et exploiter l'utilité de la visibilité des attributs et des méthodes (encapsulation).
 - Comprendre et pouvoir exploiter le polymorphisme.
- Être en mesure de programmer (et d'utiliser) des types de données abstraits (piles, files et listes) avec différentes implémentations.
- Utiliser les composants Swing pour la création d'interface graphique.

L'étudiant doit au préalable avoir une compréhension des concepts suivants^[1] :

- Types de base (Entier, réel, booléen, caractère).
- Opérateurs sur les types de base (+, -, *, /, %, ...).
- Types composés (chaînes de caractères et tableaux).
- Structures de contrôle (sélection, itération).
- Structures de programme (bloc principal, procédures et fonctions).
- Passage de paramètres (par valeur, par référence, formel et actuel).
- Entrées (clavier) et sorties (écran).

[1] Peu importe le langage de programmation qui a été utilisé.

Stratégies pédagogiques

Les deux premiers cours serviront à une révision de concepts présumés comme étant déjà connus et à l'apprentissage de ceux-ci en Java. Les cours suivants présenteront les concepts de programmation orientée-objet et des applications de ceux-ci à l'aide de notions de programmation plus approfondies et de l'utilisation des composants *Swing*.

3 heures de cours magistraux par semaine

3 heures de laboratoire par semaine

Les laboratoires visent :

- L'assimilation des notions vues au cours à l'aide d'exercices;
- La mise au point des programmes donnés en travaux pratiques (si les exercices sont finis).

Utilisation d'appareils électroniques

Micro-ordinateurs en laboratoire. Le droit à l'utilisation d'appareils électroniques en classe reste la prérogative de l'enseignant.

Coordonnées de l'enseignant

Groupe	Nom	Activité	Courriel	Local	Disponibilité
01	Pierre Bélisle	Activité de cours	Pierre.Belisle@etsmtl.ca	B-2524	
02	Pierre Bélisle	Activité de cours	Pierre.Belisle@etsmtl.ca	B-2524	
03	Frédéric Simard	Activité de cours	Frederic.Simard@etsmtl.ca	B-2568	
04	Frédéric Simard	Activité de cours	Frederic.Simard@etsmtl.ca	B-2568	

Cours

- La première période de trois heures sert aussi à présenter le plan de cours, l'approche pédagogique utilisée et les différents modèles d'évaluation.
- Notez que l'ordre de présentation peut varier selon l'enseignant.

Unité	MATIÈRE
1 - 5	<ul style="list-style-type: none">• Environnement Java.• Programmation de base en Java.<ul style="list-style-type: none">◦ Types primitifs (variables et littéraux), variable, constantes, conversion de type (<i>typecasting</i>), opérateurs (+, -, *, >=, <=, ...), instructions de contrôle (<i>if-else</i>, <i>while</i>, <i>do-while</i>), entrées/sorties (<i>Scanner()</i>, <i>System.out</i>), programme principal (<i>main()</i>).• Normes de programmation
	<ul style="list-style-type: none">• Introduction aux méthodes (procédures et fonctions), aux enregistrements (attribut à accès publique), aux classes (attribut à accès privé) et aux objets• Conception dans le paradigme orientée-objet<ul style="list-style-type: none">◦ Définitions<ul style="list-style-type: none">■ Surcharge (<i>overload</i>) et redéfinition (<i>override</i>).■ Modificateur accès (public et private)■ Constructeur■ <i>this()</i>■ Classes immuables et accesseurs (<i>get</i>),■ Classes mutables et mutateurs (<i>set</i>)■ Destructeur (<i>finalize</i>)■ Constantes <i>static</i>■ Commentaire Javadoc• Les paquetages (<i>package</i>) <i>java.lang</i> et <i>java.util</i> de Java<ul style="list-style-type: none">◦ Classes de Java : <i>String</i>, <i>Arrays</i> et <i>Math</i>• Les API de Java

	<ul style="list-style-type: none"> • Programmation de base en Java. <ul style="list-style-type: none"> ◦ Collection de base en Java <ul style="list-style-type: none"> ■ Les tableaux statiques <ul style="list-style-type: none"> ■ Boucle for ■ Tableaux <ul style="list-style-type: none"> ■ De données de type primitif ■ D'objets ◦ Algorithmes de Tris : sélection, insertion et bulles ◦ Algorithme de fouilles : linéaire, linéaire ordonnée et binaire (ou dichotomique)
	<ul style="list-style-type: none"> • Collection <ul style="list-style-type: none"> ◦ Types de données abstraits (TDA) à implémentation statique <ul style="list-style-type: none"> ■ Pile (<i>LIFO</i>) ■ File (<i>FIFO</i>) ■ Liste ◦ Levée (throws et throw) et gestion d'exception (<i>try-catch</i>).
	<ul style="list-style-type: none"> • Classe interne (<i>inner class</i>) • Classes enveloppeurs (<i>wrapper</i>) Integer, Double, ... et conversion automatique (<i>autoboxing et unboxing</i>) (java 5.0 et +). • Collection (suite) <ul style="list-style-type: none"> ◦ Chaînage dynamique simple ◦ Pile et File à implémentation dynamique
6	<ul style="list-style-type: none"> • Collection (suite) <ul style="list-style-type: none"> ◦ Liste (implémentation : chaînage dynamique simple) <ul style="list-style-type: none"> ■ Avec position numérique en paramètre (comme un tableau) <p>Examen 1 (cours 1 – 5) :</p>
7 – 9	<ul style="list-style-type: none"> • Collection (suite) <ul style="list-style-type: none"> ◦ Liste <ul style="list-style-type: none"> ■ Avec position_courante (comme un fichier) ■ Chaînage dynamique double (un exemple) • Ajout de comportement à une classe <ul style="list-style-type: none"> ◦ Par héritage (classe ou interface) <ul style="list-style-type: none"> ■ Utilité ■ Retour sur la redéfinition (override) ■ Syntaxe (extends, super) ■ Compatibilité parent-enfant ◦ Par composition • Généralisation/spécialisation (un exemple)
10	<ul style="list-style-type: none"> • Polymorphisme dynamique (<i>late binding</i>) <p>Examen 2 (cours 6-9) :</p>
11-13	<p>Les interfaces graphiques Swing sont présentées comme prétexte d'utilisation des trois concepts orientés-objet soit : Encapsulation, héritage et polymorphisme dynamique.</p> <ul style="list-style-type: none"> • GUI (Graphical User Interface) <ul style="list-style-type: none"> ◦ Composants <i>Swing</i> <ul style="list-style-type: none"> ■ JFrame, JPanel, JButton JTextField, JMenuBar, JMenu, JMenuItem, Layout Manager, ... ◦ Gestion d'évènements (bas et haut niveau).
	Examen final (cours 1-13)

Laboratoires et travaux pratiques

Laboratoires :

- Chaque semaine des exercices sont proposés pour comprendre la théorie de la semaine en vue de la réalisation des travaux pratiques. Les laboratoires ne sont pas à remettre mais nous supposons que l'étudiant les réalise. Nous pouvons les utiliser en référence dans les cours, les examens et/ou dans les travaux pratiques.

Utilisation d'outils d'ingénierie

S.O.

Évaluation

Évaluation	Pondération	Semaine	Gr. 01	Gr.02	Gr.03	Gr.04
Mini-test # 1	5 %	3	17 septembre	22 septembre	18 septembre	15 septembre
Mini-test # 2	5 %	4	24 septembre	29 septembre	23 septembre	25 septembre
Devoir # 1	10 %	5	1 octobre	6 octobre	9 octobre	6 octobre
Devoir # 2	15 %	9	29 octobre	3 novembre	30 octobre	3 novembre
Examen Intra	20 %	10	5 novembre	10 novembre	10 novembre	12 novembre
Travail # 3	15 %	13	6 décembre	6 décembre	4 décembre	1 décembre
Examen final (3h00)	30 %	Période des examens finaux				

À propos des travaux pratiques :

- Les travaux pratiques visent à mettre en application, dans le cadre d'un problème réaliste, les concepts et les méthodes de programmation enseignés pendant les cours théoriques. Il est attendu des étudiant-e-s de produire des travaux qui permettent de résoudre le problème de l'énoncé MAIS SURTOUT de les réaliser selon les standards de qualité et les bonnes pratiques enseignés (respect des normes, décomposition en sous-programme/modules, etc.). La cohérence et l'optimalité des algorithmes implémentés constituent également des critères importants qui seront considérés. En tout état de cause, la note attribuée à l'exécution correcte du programme ne pourra excéder **40%** de la note globale d'un travail pratique.
- Pour chaque travail pratique, l'enseignant indiquera sur l'énoncé s'il doit être réalisé en équipe et, si tel est le cas, la taille de l'équipe. Une équipe ne doit remettre qu'un seul rapport.
- Il est à noter que pour les travaux en équipe, chaque membre recevra une note inférieure ou égale à la note obtenue par l'équipe. Cette note individuelle est directement proportionnelle à l'implication de l'individu. La méthode d'évaluation de l'implication de chaque individu est laissée à la discrétion de l'enseignant.

Double seuil :

- Une note moyenne pondérée de 50 % est exigée pour l'ensemble des évaluations à caractère individuel. Ce seuil est une condition nécessaire à la réussite du cours mais ne la garantit pas.

Examens :

- Aucune documentation n'est permise.
- L'utilisation de la calculatrice est interdite.

Dates des examens intra

Groupe(s)	Date
1	5 novembre 2020
2, 3	10 novembre 2020
4	12 novembre 2020

Date de l'examen final

Votre examen final aura lieu pendant la période des examens finaux, veuillez consulter l'horaire à l'adresse suivante : <http://etsmtl.ca/Etudiants-actuels/Baccalaureat/Examens-finaux>

Politique de retard des travaux

Tout travail (devoir pratique, rapport de laboratoire, rapport de projet, etc.) remis en retard sans motif valable, c'est-à-dire autre que ceux mentionnés dans le Règlement des études (1er cycle, article 7.2.7 b / cycles supérieurs, article 6.5.4 b) se verra attribuer la note zéro, à moins que d'autres dispositions ne soient communiquées par écrit par l'enseignant dans les consignes de chaque travail à remettre ou dans le plan de cours pour l'ensemble des travaux.

Absence à un examen

Dans les cinq (5) jours ouvrables suivants, 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 ou du SEG. Pour un examen final, l'étudiant devra justifier son absence auprès du Bureau du registraire. Dans tous les cas, l'étudiant doit effectuer sa demande en complétant le formulaire prévu à cet effet qui se trouve dans son portail Mon ÉTS/Formulaires. Toute absence non justifiée par un motif majeur (maladie certifiée par un billet de médecin, décès d'un parent immédiat, Activité compétitive d'un étudiant appartenant à un club scientifique ou un club sportif d'élite de l'ÉTS ou au programme « Alliance sport étude » ou autre) à un examen entraînera l'attribution de la note zéro (0).

Plagiat et fraude

Les clauses du « Règlement sur les infractions de nature académique de l'ÉTS » s'appliquent dans ce cours ainsi que dans tous les cours du département. Les étudiants doivent consulter le [Règlement sur les infractions de nature académique](https://www.etsmtl.ca/docs/ETS/Gouvernance/Secretariat-general/Cadre-reglementaire/Documents/Infractions-nature-academique) (<https://www.etsmtl.ca/docs/ETS/Gouvernance/Secretariat-general/Cadre-reglementaire/Documents/Infractions-nature-academique>) pour identifier les actes considérés comme étant des infractions de nature académique ainsi que prendre connaissance des sanctions prévues à cet effet. À l'ÉTS, le respect de la propriété intellectuelle est une valeur essentielle et les étudiants sont invités à consulter la page [Citer, pas plagier !](https://www.etsmtl.ca/Etudiants-actuels/Baccalaureat/Citer-pas-plagier) (<https://www.etsmtl.ca/Etudiants-actuels/Baccalaureat/Citer-pas-plagier>).

Documentation obligatoire

Aucune référence obligatoire

Ouvrages de références

Autres références

Horstmann, Cay, *Big Java*, 3rd Edition, San Jose State Univ., John Wiley & sons, 2007, 1248 pages.

Horton's, Ivor, *Ivor Horton's beginning Java™ 2, JDK™5 Edition*, Indianapolis, Wiley Publishing, Inc., 2005, 1470 pages.

Adresse internet du site de cours et autres liens utiles

<https://ena.etsmtl.ca/course/view.php?id=537>
[Java version 8](#)

