

IFT 2255 – GÉNIE LOGICIEL – A18

DESCRIPTION :

Introduction au génie logiciel. Cycles de développement. Analyse, modélisation et spécification. Conception. Développement orienté objet. Mise au point. Outils et environnements de développement.

CHAPITRES :

1. Développement
2. Exigences et analyse
3. Conception
4. Implémentation
5. Évolution

LIENS RAPIDES

- <https://admission.umontreal.ca/cours-et-horaires/cours/IFT-2255/>
- <https://michalis.famelis.info/teaching-2/ift-2255-genie-logiciel-automne-2018/>
- <https://ift2255-a18.slack.com/>
- <https://studium.umontreal.ca/course/view.php?id=129169>

ENSEIGNANTS :

Michalis Famelis, chargé de cours

<http://www-ens.iro.umontreal.ca/~famelis/>

Bureau: Pav. André-Aisenstadt local 2343

Disponibilité au bureau sans rendez-vous : mardi 16:30-17:15

Maude Sabourin, auxiliaire

Thomas Schweitzer, auxiliaire

Disponibilités : sur rendez-vous

HORAIRES :

Cours : local S-144 Pav. Roger-Gaudry

- Mardi – 14:30 - 16:30
- Vendredi – 13:30 - 14:30

Démonstrations : local S-118 Pav. Roger-Gaudry

- Vendredi – 14:30 - 16:30

Examens :

- Intra : vendredi 2 novembre, 14:30-16:20, local 1140 Pav. André-Aisenstadt
- Final : vendredi 14 décembre, 14:30-17:20, local 1140 Pav. André-Aisenstadt

COMMUNICATIONS :

FORUM :

Nous utiliserons Slack comme le principal moyen de communication et discussion :

<https://ift2255-a18.slack.com/>

Annonces affichées sur le canal *#general* sur Slack, seront supposées avoir été vues par tous les étudiants.

POLITIQUE :

- Pour toutes les communications concernant le cours, utilisez Slack. Pour tout autre problème et pour les demandes confidentielles, contactez-moi à famelis@iro.umontreal.ca.
- Nous allons essayer de répondre à tous vos messages avant la fin du prochain jour ouvrable. Nous ne pouvons pas garantir de répondre en dehors des heures de bureau ou pendant les fins-de-semaine. Nous ne nous attendons pas non plus à ce que vous répondiez pendant ces périodes.

ÉVALUATION

Devoirs maison : 40% (pondérés selon la difficulté)

Examen intra : 20%

Examen final : 40%

Les devoirs maison vous initieront au développement logiciel collaboratif :

- 3 devoirs (~3 semaines chaque) en groupe
- Le langage de programmation utilisé sera Java quand nécessaire
- Tous les travaux seront évalués selon (dans l'ordre): l'exactitude, la qualité du travail, la structure et l'état complet, la présentation
- Soumission: tous les devoirs sont dus à la date correspondante avant 23:55
- Il y a 5% de pénalité par jour en retard, jusqu'à 2 jours au maximum, après quoi le devoir sera refusé

Il n'est pas possible de refaire ou faire plus de travail pour augmenter sa note.

LIVRE DU COURS

Object-oriented and classical software engineering, par Stephen R. Schach, Boston ; Montreal : McGraw-Hill c2002
QA 76.758 S33 2002

UML 2 : De l'apprentissage à la pratique, Laurent Audibert
<https://laurent-audibert.developpez.com/Cours-UML/>

(Optionnel) *Software engineering : the current practice*, Václav Rajlich, Boca Raton, FL : CRC Press c2012
QA 76.758 R35 2012

(La bibliothèque des mathématiques et informatique a une très bonne collection des livres sur le sujet. Vous êtes bien invités à la feuilleter.)

NOTEZ BIEN

Présence en cours : les diapos sont peu nombreuses, donc une présence aux cours est indispensable pour une bonne compréhension de la matière.

Plagiat : les discussions avec les autres étudiants du cours sont encouragées, mais le plagiat sous toute forme sera puni, conformément aux règlements de l'Université de Montréal (échec du cours, sanctions).

<http://integrite.umontreal.ca/>

Avertissement : ce document n'est qu'informatif et pourrait différer du matériel, supports et communications sont disponibles dans StudiUM et Slack. En cas de divergence, les annonces sur StudiUM et Slack doivent être supposées correctes. Les étudiants sont donc obligés à suivre ces canaux de communication (spécialement le canal #general sur Slack) attentivement.

Langue : je suis aussi un étudiant de français. Veuillez patienter avec mon français – jusqu' à ce que j'atteigne un niveau plus sérieux.

PLAN DE COURS (PRÉVISIONNEL) :

Chapitre		Théorie @ S-144	Démo @ S-118	Devoirs	Lectures
Logiciel	4 sept.	Introduction + Vie du logiciel			ch.1, Rajlich ch.1
	7 sept.	Modèles de développement	<i>Pas de démo</i>		ch.2, Rajlich ch.2-3
	11 sept.	Processus de développement			ch.3
	14 sept.	Logiciel en équipe + Versionnement de logiciel	Équipes + Outils nécessaires		ch.4, 5.9-5.11, Rajlich ch.3.3
Exigences & Analyse	18 sept.	Exigences			ch.11-11.4
	21 sept.	Cas d'utilisation	Contrôle de révisions + Besoins	DM1: Création	ch.11.5-11.18
	25 sept.	Analyse			ch.17.4,17.5
	28 sept.	Analyse	Diagrammes de cas d'utilisation + activités		ch.13
Conception	2 oct.	Conception			ch.14
	5 oct.	Modélisation	Diagrammes de classes participantes		UML UG ch.18
	9 oct.	Modélisation			ch.17, UML UG ch.4,5,8,9
	12 oct.	Architecture	(Théorie:) Architecture	DM1 dû	ch.17, UML UG ch.18
	16 oct.	(Démo:) Correction DM1 + Modélisation		DM2: Élaboration	ch.7-7.3, 8-8.5.2
Implémentation	19 oct.	(Aux:) Paradigme orienté objet	Diagrammes de séquence		ch.8.11-8.13
	23 oct.	<i>Relâche</i>			
	26 oct.	<i>Relâche</i>			
	30 oct.	Révision intra	Architecture + Patrons		ch.7.4-7.9
	2 nov.	Examen intra 14:30-16:20 @ 1140 Pav. André-Aisenstadt			
	6 nov.	Patrons de conception			ch.8.6-8.8
	9 nov.	Patrons de conception	Correction intra + Support DM2		
	13 nov.	Implémentation		DM2 dû	ch.15
	16 nov.	Vérification et validation	Correction DM2 + Implémentation	DM3: Construction	ch.6,15.18-15.23
	20 nov.	Vérification et validation			ch.15.9-15.18
	23 nov.	Tests unitaires	JUnit		ch.16
Évolution	27 nov.	Déploiement + Maintenance			Rajlich ch.5-7
	30 nov.	Workflow de maintenance	Support DM3		Rajlich ch.8-11
	4 déc.	Workflow de maintenance + License et éthique			
	7 déc.	Conclusion	Maintenance + Révision	DM3 dû	
	14 déc.	Examen final 14:30-17:20 @ 1140 Pav. André-Aisenstadt			