

COORDONNATEUR	SALAH, Aziz	salah.aziz@uqam.ca	(514) 987-3000 1485	PK-4835
GROUPE	10 MALENFANT, Bruno	malenfant.bruno@uqam.ca	(514) 987-3000 4005	PK-4635

Lundi, de 13h30 à 15h00 et de 15h15 à 16h45 (cours) – Mardi, de 14h00 à 16h00 (ateliers)

**DESCRIPTION**

Initier les étudiants à la programmation à l'aide d'un langage impératif et procédural. Familiariser les étudiants à la construction professionnelle de logiciels et à leur maintenance.

Notions de base de la programmation procédurale et impérative en langage C sous environnement Unix/Linux (définition et déclaration, portée et durée de vie, fichier d'interface, structures de contrôle, unités de programme et passage des paramètres, macros, compilation conditionnelle). Décomposition en modules et caractéristiques facilitant les modifications (cohésion et couplage, encapsulation et dissimulation de l'information, décomposition fonctionnelle). Style de programmation (conventions, documentation interne, gabarits). Débogage de programmes (erreurs typiques, traces, outils, par ex., gdb). Assertions et conception par contrats. Tests (unitaires, intégration, d'acceptation, boîte noire vs. boîte blanche, mesures de couverture, outils d'exécution automatique des tests, par exemple, xUnit, scripts). Évaluation et amélioration des performances (profils d'exécution, améliorations asymptotiques vs. optimisations, outils). Techniques et outils de base pour la gestion de la configuration (par exemple, make, cvs). Introduction à la maintenance de logiciels (types de maintenance, techniques de base, par exemple, remodelage, automatisation des tests de régression). Ce cours comporte une séance obligatoire de laboratoire (2 heures).

Préalables: INF2120 Programmation II

**OBJECTIFS**

Ce cours vise à introduire les étudiants à la construction professionnelle de logiciels. Il vise aussi à familiariser les étudiants avec la programmation procédurale et impérative.

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

- de développer et modifier des composants logiciels écrits dans un langage impératif et procédural;
- de bien maîtriser le langage C et le compilateur du C sous UNIX/Linux;
- d'utiliser les notions de module, de cohésion et couplage, de complexité structurale, de dissimulation de l'information, etc., pour évaluer la qualité d'un composant logiciel;
- d'expliquer et d'utiliser les principales techniques de modularisation : décomposition fonctionnelle, dissimulation de l'information, filtres et pipelines;
- d'utiliser des assertions (pré/post-conditions, invariants) pour documenter des composants logiciels et assurer leur bon fonctionnement;
- de déboguer un programme à l'aide de techniques, stratégies et outils appropriés;
- de vérifier le bon fonctionnement d'un composant logiciel à l'aide de tests fonctionnels et structurels, et d'évaluer à l'aide des notions et outils appropriés la qualité des tests (par ex., complexité cyclomatique, mesures de couvertures des tests);
- d'évaluer de façon empirique à l'aide d'outils appropriés (par ex., profils d'exécution) les performances d'un composant logiciel de façon à pouvoir, si nécessaire, en améliorer les performances;
- d'utiliser divers outils (outil de gestion de configuration, fichiers makefile, langage de scripts) pour organiser le développement de programmes comportant plusieurs composants ou modules;
- d'expliquer les notions de base de la maintenance des logiciels et d'appliquer certaines techniques de maintenance (tests de régression exécutés automatiquement, remodelage de programmes).

ÉVALUATION	Description sommaire	Date	Pondération
	Examen intra	8ième semaine	30%
	Examen final	15ième semaine	30%
	TP 1		10%
	TP 2, 3		15% chacun

Toute documentation personnelle est permise aux examens. Une moyenne d'au moins 50% aux examens est exigée pour réussir le cours.

Les travaux pratiques peuvent être réalisés individuellement ou en équipe de deux (2) personnes. Par contre, les examens doivent évidemment être faits de façon individuelle. Une pénalité de 10% par jour de retard sera appliquée pour la remise des travaux pratiques. La qualité du français sera prise en considération (jusqu'à 10 % de pénalité).

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html> et <http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

### Politique d'absence aux examens

**L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.**

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen – par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : SH-4700 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

### CONTENU

#### Contenu à titre indicatif

- 1 Présentation du plan de cours. Conception.
- 2 Langage C : présentation générale, noms des variables et fonctions, instruction for, variables globales et visibilité.
- 3 Langage C (suite) : types de base, conversions, expressions et priorités des opérateurs, expressions conditionnelles, boucles, fonction main. Style de programmation.
- 4 Langage C (suite) : variables externes et statiques, préprocesseur, pointeurs et tableaux. Débogage.
- 5 Langage C (suite) : tableaux multi-dimensionnels, allocation dynamique de mémoire, arguments du programme.
- 6 Langage C (suite) : structures, typedef, union.
- 7 Langage C (suite) : listes chaînées, fichiers et entrées/sorties standards. Révision pour l'examen Intra.
- 8 Examen intra.
- 9 Gestion de la configuration : cvs, make. Outils Unix (scripts).
- 10 Programmation par contrats. Programmation défensive
- 11 Tests.
- 12 Tests (suite). Maintenance.
- 13 Évaluation des performances. Révision pour l'examen final.
- 14 Examen final.

### RÉFÉRENCES

- VC Kernighan, B.W. et Ritchie, D.M. – *Le langage C – deuxième édition*, Masson, Paris, 1990.
- VC Blaquelaire, J.P. – *Méthodologie de la programmation en C (3e édition)* – Masson, 1998. [QA76.73 C15B75].
- VC Kernighan, B.W. et R. Pike, R. – *The Practice of Programming* – Addison-Wesley, 1999.
- VC Kernighan, B.W. et R. Pike, R. – *La programmation - En pratique* – Vuibert, 2001. [QA76.6K48814].
- VC Loukides, M. et Oram, A. – *Programming with GNU Software* – O'Reilly, 1997.
- VC McConnell, S. – *Code Complete - A Practical Handbook of Software Construction – (Second edition)*. Microsoft Press, Redmond, WA, 2004.

---

<sup>VC</sup> Purdy, G.N. – *CVS Précis & concis* – Éditions O'Reilly, 2004.

<sup>VC</sup> ZELLER, A. and KRINKE, J. – *Essential Open Source Toolset* – John Wiley & Sons, Ltd, 2005.

<sup>UR</sup> <http://www.info2.uqam.ca/~tremblay/INF3135>

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –  
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé