



Université du Québec (UQ)

**École de technologie supérieure**

Service des enseignements généraux

Local B-2500 – (514) 396-8938

Site internet : <http://www.seg.etsmtl.ca>

## PLAN DE COURS

### INF-145 PROGRAMMATION AVANCÉE ET LANGAGE C

HIVER 2010

Enseignant :	<a href="#">Huques Saulnier</a> (gr. 01, 02 et 03)	local B-2306
Coordonnateurs :	<a href="#">Pierre Bélisle</a> <a href="#">Frédéric Henri</a>	local B-2524 local B-2514

## OBJECTIFS GÉNÉRAUX

Ce cours a pour objectif de procurer une connaissance moderne de la programmation, pour qu'un étudiant puisse solutionner des problèmes liés à sa discipline.

Le langage de programmation utilisé est le C ANSI 99.

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

- D'acquérir les notions de programmation de base
- D'acquérir une connaissance correcte du langage C
- D'utiliser et d'implémenter des types de données abstraits

## STRATÉGIES PÉDAGOGIQUES

Nous utiliserons principalement le compilateur Visual C++ en évitant soigneusement toute librairie et toute fonction qui ne soient pas ANSI, en mettant l'accent sur la portabilité du code.

3 heures et demie de cours par semaine.

3 heures de laboratoire par semaine.

Les laboratoires visent à :

- L'assimilation par l'expérience des notions vues au cours
- La mise au point de programmes donnés en travaux pratiques.

La séance de laboratoire pourra être jumelée avec celle du cours de façon à ce que chaque rencontre hebdomadaire se compose d'une période de théorie et d'une période de laboratoire.

## HORAIRE DES COURS

Ce tableau est donné à titre indicatif. Il se peut que votre enseignant modifie l'ordre de présentation des notions pour des raisons pédagogiques.

COURS	MATIÈRE
1 à 7	<p>Description du fonctionnement du cours et des différentes ressources disponibles à l'étudiant (enseignants, livres, site web).</p> <p>Présentation des notions élémentaires de programmation : types de base du C avec leurs opérateurs et les conversions automatiques, variables, entrées/sorties de base, instructions conditionnelles et itératives et <i>l'algorithmie de base</i>.</p> <p>Description de la qualité de programmation attendue : commentaires, indentation, utilisation de constantes et qualité des identificateurs.</p> <p>Présentation de commandes au préprocesseur (<code>#define #include</code> et <code>#if-#endif</code>)</p> <p>Fonctions : déclaration, commentaires nécessaires à la déclaration, paramètres effectifs et paramètres formels, valeur de retour, passage par valeur, portée et durée de vie des variables, découpage et modules de fonctions.</p> <p>Tableaux : déclaration, initialisation, accès aux éléments, passage en paramètre, tableaux à plusieurs dimensions et algorithmes de base.</p> <p>Déclarations de types (<i>typedef</i>).</p> <p>Fouilles et tris.</p> <p>Fonctionnement des adresses : obtention par opérateur ou par identificateur avec l'arithmétique des adresses (lorsqu'elle est possible), l'indirection. Fonctionnement des pointeurs : déclaration, initialisation et assignation, parasitage et allocation programmée, passage par adresse, pointeurs génériques. Classes d'allocation.</p> <p>Chaînes de caractères : chaînes littérales et variables de type « string », les multiples bibliothèques et fonctions offertes par le langage pour leur traitement.</p> <p>Lecture de déclarations C plus complexes impliquant des pointeurs.</p>
8 à 13	<p>Enregistrements : déclaration, accès aux membres et passage en paramètre.</p> <p>Modules de définition de type avec les fonctions nécessaires à une construction bien faite.</p> <p>Représentation cachée : encapsulation de données comme du traitement avec le concept de descripteur (« <i>handle</i> »).</p> <p>Fichiers texte et binaire : le <code>File*</code> et ses fonctions classiques.</p> <p>Introduction aux structures de données.</p> <p>Piles et files : intérêt, fonctionnement et implémentations sur tableaux</p> <p>Chaînage dynamique : concept de nœud et de lien.</p> <p>Liste : intérêt, fonctionnement et implémentations.</p>

## ÉVALUATIONS

Ce tableau est donné à titre **indicatif**. En cours de session, votre enseignant vous fera part à l'avance de tout changement à l'horaire d'évaluation. Un horaire précis sera maintenu sur la page Web.

Évaluation	Semaine
Test numéro I (11%)	3
Test numéro II (11%)	5
Remise du devoir (13%)	À déterminer
Test numéro III (11%)	7
Remise du travail pratique I (13%)	À déterminer
Test numéro IV (14%)	À déterminer
Test numéro V (14%)	13
Remise du travail pratique II (13%)	Semaine 1 des examens finaux

### À propos des tests :

Un test va durer de 50 à 70 minutes et aucune documentation n'est permise. Toute absence non justifiée par un motif majeur (maladie avec billet de médecin, accident, décès malheureux d'un proche ou autre triste événement), entraîne automatiquement l'attribution de la note « 0 » zéro. L'étudiant devra démontrer que son absence était justifiée.

### À propos des travaux pratiques :

- Les devoirs et travaux ont pour but de mettre en pratique les connaissances acquises pendant les cours théoriques. Pour chacun d'eux, 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.
- La présentation du dernier travail pratique sera faite lors d'une rencontre avec l'enseignant par tous les membres de l'équipe durant la période des examens finaux. Toute absence d'un membre de l'équipe non justifiée par un motif majeur entraîne l'attribution automatique de la note 0 à cet étudiant.
- 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 à l'ensemble du développement et à la bonne connaissance des modules offerts.

## PRINCIPALES RÉFÉRENCES

Le choix d'un livre est laissé à la discrétion de l'étudiant. Les livres portant sur le langage C énumérés ci-dessous sont tous jugés comme étant adéquat pour ce cours. Voici quelques livres portant sur le langage C et certaines références diverses plus spécialisées

MILLER, Lawrence H. & Alexander E. QUILICI. *The Joy of C – Programming in C*, Wiley, 1997.

*Excellent manuel sur le C complet. Rien n'est laissé au hasard. Tous les étudiants auraient avantage à connaître ce bouquin. Très longuement utilisé dans le cours INF125.*

DRIX, Philippe. *Le langage C ANSI – Vers une pensée objet en Java*, 3<sup>e</sup> édition, InterEditions, 1993.

*Ce livre s'adresse à un lecteur connaissant déjà la programmation (préférentiellement en Pascal) et l'amène aux portes de la programmation orientée objet. Son approche et ses objectifs correspondent très bien à la mentalité du cours. Son seul défaut est qu'il n'aborde pas les structures de données.*

HARBISON III, Samuel P. & Guy L STEELE JR. *C – A reference manual*, 5e edition, Prentice Hall, 2002.

*Ce livre décrit le langage C et un style de programmation axé sur la portabilité et la maintenance. Il ne se veut pas un guide d'apprentissage de la programmation en C. En plus d'être extrêmement précis, il discute de la norme C99, ce qui est rare dans les livres de C pour l'instant.*

BRASSARD, Gilles & Paul BRATLEY. *Algorithmique – Conception et analyse*, Masson, 1987.

*Un manuel avancé dépassant largement les objectifs de notre cours. Son intérêt se situe au niveau des algorithmes touchant les arbres et les graphes. Si vous souhaitez approfondir ces sujets ou étudier la notion d'algorithmie, alors ce livre devrait vous intéresser.*

KOENIG, Andrew. *C traps and pitfalls*, 5<sup>e</sup> édition, Addison-Wesley, 1989.

*Un tout petit livre qui aborde les embûches courantes rencontrées par le programmeur en C. Bien que notre cours dépasse largement les notions présentées dans ce livre, celui-ci recèle des informations que tout étudiant doit savoir à la fin du cours.*