



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

Automne 2009

Enseignant : [Frédéric Henri](#) (gr. 01 & 02) local B-2514

Coordonnateurs : [Frédéric Henri](#) local B-2514  
[Pierre Bélisle](#) local B-2524

## OBJECTIFS GÉNÉRAUX

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

Le langage de programmation utilisé est le C ANSI 89.

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

- acquérir les notions de programmation de base;
- acquérir une connaissance du langage C ;
- utiliser et implémenter des types de données abstraits (pile, file et liste).

## STRATÉGIES PÉDAGOGIQUES

Nous utiliserons principalement le compilateur Visual C++ en évitant soigneusement toute librairie et toute fonction qui ne soient pas ANSI et 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 :

- 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	<p><b>Notions élémentaires de programmation</b> : types numériques, opérateurs, variables, entrées/sorties de base, instructions conditionnelles et itératives.</p> <p><b>Qualité de programmation attendue</b> : commentaires, indentation, utilisation de constantes et nomenclature des identificateurs.</p>
2	<p><b>Commandes du préprocesseur en C</b> : <code>#define</code> (constantes de compilation et macros-fonctions), <code>#include</code> et <code>#if - #endif</code>.</p> <p><b>Types arithmétiques</b> : opérateurs et conversion de type.</p>
3	<p><b>Fonctions</b> : déclaration, définition, commentaires de l'en-tête, paramètres effectifs et paramètres formels, valeur de retour, passage par valeur, portée et durée de vie des variables.</p>
4	<p><b>Pointeurs</b> : déclaration, opérations permises (arithmétique des pointeurs et indirection), passage par adresse et pointeurs constants.</p> <p><b>Tableaux</b> : déclaration, initialisation, accès aux éléments, opérations permises, passage en paramètre et tableaux de tableaux.</p> <p><b>Définition de types</b> : <code>typedef</code>.</p>
5	<p><b>Algorithmes sur tableaux</b> : fouilles (linéaire et dichotomique) et tris (par sélection, par insertion et bulle).</p> <p><b>Chaînes de caractères</b> : type <code>char</code>, déclaration, fonctionnement, opérations permises, chaînes littérales, fonctions offertes par le langage et entrées/sorties.</p>
6	<p><b>Classes d'allocation</b> : <code>auto</code>, <code>register</code>, <code>static</code> et <code>extern</code>.</p> <p><b>Pointeurs et adresses</b> : obtention d'adresses, pointeurs génériques (<code>void *</code>) et allocation programmée de mémoire.</p>
7	<b>EXAMEN I</b>
8	<p><b>Types étiquetés</b> : <code>struct</code> et <code>enum</code>.</p> <p><b>Enregistrements</b> : déclaration, accès aux champs, opérations permises et passage en paramètre.</p> <p><b>Enregistrements et allocation programmée</b> : copie de surface vs copie en profondeur.</p>
9	<p><b>Fichiers</b> : ouverture/création, entrées/sorties, fichier texte vs fichier binaire.</p> <p><b>Gestion des erreurs</b> : retour de fonction, arrêt du programme (<code>assert</code>), ignorance de l'erreur et retour à l'aide d'une variable externe.</p> <p><b>Modules</b> : de fonctions (librairie) et de définition de type.</p>

10 – 12	<p><b>Représentation cachée</b> : implémentation en C.</p> <p><b>Structures de données</b> : intérêt.</p> <p><b>Chaînage dynamique</b> : concepts de nœud et de lien.</p> <p><b>Listes chaînées</b> : intérêt, fonctionnement et implémentations.</p> <p><b>Piles et files</b> : intérêt, fonctionnement, implémentations sur tableau et à l'aide d'une liste chaînée.</p>
13	<p><u>Notions facultatives</u> :</p> <p><b>Arbres binaires et arbres binaires de fouille</b> : intérêt, fonctionnement, parcours et implantations diverses (sur tableau et avec des nœuds alloués dynamiquement).</p> <p><b>Pointeurs de fonction.</b></p> <p><b>Algorithmes récursifs.</b></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.

Évaluation	Pondération	Semaine	Gr. 01	Gr. 02
Laboratoire sur les fonctions	5 %	3	21 septembre	29 septembre
Travail #1	10 %	4	23 septembre	1 <sup>er</sup> octobre
Laboratoire sur les tableaux	5 %	5	5 octobre	6 octobre
Examen I	25 %	7	14 octobre	21 octobre
Travail # 2	20 %	10	11 novembre	12 novembre
Travail # 3	10 %	13	4 décembre	4 décembre
Examen II	25 %	Finaux	À déterminer	À déterminer

### À propos des examens :

Toute absence à un examen non justifiée par un motif majeur (maladie avec billet de médecin, accident, décès d'un proche ou autre), entraînera l'attribution de la note « 0 » (zéro) pour cet examen. L'étudiant devra démontrer que son absence était justifiée. S'il s'agit d'une absence à l'examen final, l'étudiant devra se présenter au Service de la gestion académique, local B-2700. S'il s'agit d'une absence à un autre examen, l'étudiant devra prendre entente avec son professeur et se présenter au secrétariat du Service des enseignements généraux, local B-2500, pour remplir le formulaire approprié.

## À propos des travaux pratiques :

- Les 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 peut 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. La méthode d'évaluation de l'implication de chaque individu est laissée à la discrétion de l'enseignant.

## PRINCIPALES RÉFÉRENCES

Le choix d'un livre est laissé à la discrétion de l'étudiant. Voici quelques livres portant sur le langage C et diverses références plus spécialisées

### MANUEL RECOMMANDÉ :

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.*

### RÉFÉRENCES PLUS SPÉCIALISÉES :

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.*