

# INF3105 – Introduction

Jaël Champagne Gareau

Université du Québec à Montréal (UQAM)

Été 2024

<http://cria2.uqam.ca/INF3105/>



# Sommaire

- 1 Introduction
- 2 Méthodologie d'enseignement
- 3 Évaluation
- 4 Types abstraits de données
- 5 Génie logiciel
- 6 ...

# Description d'INF3105

- Approfondir les connaissances des structures de données et des algorithmes, et les appliquer à la résolution de problèmes.
- Rappels sur les types abstraits de données. Analyse et complexité des algorithmes. Abstractions de données et de contrôle. Collections et les structures de données nécessaires à leurs réalisations. Arbres équilibrés, tables de hachage, graphes.
- Bibliothèques publiques ou normalisées. Ce cours comporte une séance obligatoire de laboratoire (2 heures).
- Préalables : INF1132 Mathématiques pour l'informatique ou MAT1060 Mathématiques algorithmiques ; INF2120 Programmation II

# INF3105 $\Rightarrow$ cours difficile

- Le sujet des **structures de données** est difficile.
- **C++  $\Rightarrow$  nouveau langage** pour la majorité d'entre vous.
- Le C++ est un langage plus difficile à maîtriser que Java.
  - Gestion de la mémoire.
  - Pointeurs et références.
  - Messages d'erreur avec les classes génériques (*templates*).
  - Etc.
- Résolution de problèmes.
- Cours fondamental et important dans la formation d'un(e) informaticien(ne).
- Théorie + Pratique.

# Théorie vs Pratique

- Certains trouveront le cours trop théorique...
- D'autres le trouveront trop pratique...

# Méthodologie d'enseignement

- Cours magistraux (ou capsules vidéos).
- Lectures personnelles :
  - Notes de cours (inf3105-notes.pdf sur le site Web du cours).
  - Un livre sur les structures de données : [Goodrich] ou autre.
  - Documentation technique sur C++ (un livre ou Web).
- Laboratoires.
- Tout est sujet à examen.

# Orientations

- Première partie : conception de nos propres structures.
- Deuxième partie : utilisation d'une bibliothèque normalisée.
- Le tout : théorique et appliqué à la résolution de problèmes.

# Contenu et Calendrier

- Ébauche : <http://cria2.uqam.ca/INF3105/>



## Suite à INF2120 (Programmation II)

- Il y aura quelques rappels d'éléments de INF2120.
- Exemples : tableau, pile, file et listes.
- C'est volontaire.
  - Nous reverrons cette matière pour vous familiariser avec C++, les pointeurs et la gestion de la mémoire.
  - Java cache certains détails qui sont pertinents à connaître d'un point de vue théorique.
  - Nous irons plus en profondeur (ex. : itérateurs de liste).

# Politiques sur les outils de développement

- Liberté dans le choix des environnements de développement.
- Compilateur C++ supportant la norme ISO de 2011 («C++11»).
- Nous n'utiliserons qu'une petite partie de la bibliothèque standard de C++.
- Obligation : compilation avec GNU GCC (g++) sous Ubuntu Linux 22.04.
- g++ version 12.
- Environnement recommandé : l'environnement minimaliste.
- Pas besoin d'un EDI (environnement de développement intégré).
- Voir <http://cria2.uqam.ca/INF3105/politiques/>

# Laboratoires

- Organisation typique d'un lab :
  - Courte présentation par le démonstrateur.
  - Orienté sur une structure de donnée vue au cours précédent.
  - Téléchargement d'un squelette de départ.
  - Vous complétez des morceaux laissés en exercice.
  - Les démonstrateurs répondent à vos questions.
  - Vous exploitez la structure développée pour résoudre un petit problème précis.
  - Le démonstrateur fait un résumé si nécessaire.
- Les structures développées dans les labs sont souvent réutilisables dans les TPs.
- Certaines tâches facultatives (dépassement des 2 heures prévues).

# Charge de travail

- Officiellement : 2 heures de travail personnel par heure de cours.
- La session dure 15 semaines : 3 heures de cours + 6 heures de travail personnel = 9 heures / semaine.
- Total : 90 heures de travail personnel.
- Enquête volontaire (sondage non scientifique) auprès d'étudiants ayant fait le cours :
  - 20 % :  $\leq$  60 heures pour la session.
  - 60 % : 60 à 150 heures pour la session.
  - 20 % :  $>$  150 heures pour la session.
- **Si vous avez l'impression que le cours est trop difficile ou que vous y consacrez trop de temps, consultez-moi.**

# Généralités sur l'évaluation

- Critères généraux : **correct**, **efficacité**, lisibilité, robustesse, etc.
- Quiz de lecture (5e et 13e semaines)
- Examens :
  - ~~Aucune documentation permise.~~
  - Un aide mémoire sur C++ est parfois fourni.
  - Questions théoriques et écriture de code.
  - Connaissance de C++.
- Travaux pratiques :
  - 5 Travaux pratiques.
  - Chaque TP a un objectif ciblé.
  - Remises électroniques.

# Évaluations en ligne et Intégrité académique

- Défis des évaluations en ligne : identité, aide, plagiat, etc.
- Vous aurez à produire quelques vidéos où vous expliquerez vos démarches et réponses.
- Vous aurez besoin de matériel pour enregistrer vidéo et audio (webcam, téléphone intelligent, caméra, etc.).
- Détails à venir.

# Politique sur les travaux pratiques en équipe

- Taille des équipes : 1 ou 2 (sauf le TP5 devant être individuel).
- Le travail en équipe a pour but de vous entraider.
- Le travail en équipe n'a pas pour but de répartir la charge de travail afin de la diviser par deux.
- **Tous les membres de l'équipe doivent participer à toutes les tâches.**
- Bonne méthode de travail : «extreme programming» (2 devant le même écran).
- Mauvaise méthode de travail : travailler chacun de son côté sur des parties différentes.
- Mauvaise méthode de travail = plagiat.
- En cas de doute, je convoque l'équipe pour une évaluation orale.

# Politique sur le report d'une date de remise

- Si vous avez besoin de plus de temps pour un TP, avertissez-moi avant le début du cours qui précède la remise du TP.
- Donc au moins une semaine à l'avance.
- Si je juge nécessaire d'accorder plus de temps (je me garde un droit de veto), je peux proposer de reculer la date de remise.
- Cette proposition doit être adoptée à la majorité en classe.



# Pénalité pour remise en retard

- Pénalité linéaire de 5 % de la note maximale par heure de retard.
- Après 20 heures de retard, la note est automatiquement zéro.
- Les retards sont calculés à partir de l'heure de remise.
- L'heure officielle du cours est celle de l'horloge du Conseil national de recherches Canada (<https://nrc.canada.ca/fr/horloge-web/>) !

# Politique d'absence aux examens

- Reprise possible si motif sérieux (maladie, décès proche, etc.).
- Un conflit d'horaire, même sur examen, n'est pas considéré comme un motif sérieux.
- En tant qu'enseignant de INF3105, je ne suis pas autorisé à accepter ou à négocier un arrangement pour les examens. Donc, inutile de venir me voir, de m'envoyer un billet médical ou certificat de décès d'un proche, etc. Vous devez voir avec la direction de programme.
- Présentez votre demande à <https://info.uqam.ca/repriseexamen/>

# Politique d'absence aux quiz

- Aucune reprise en cas d'absence (qu'elle soit motivée ou non).
- Aucune absence :  $NoteQuiz = \max(quiz1, quiz2)$ .
- En cas d'une seule absence motivée :  $NoteQuiz = \max(quiz1, quiz2)$ .
- En cas de deux absences motivées :  $NoteQuiz = 5 \cdot \alpha(exam1 + exam2)$ .
- où  $\alpha = \frac{\mu_{quiz}}{\mu_{exam}}$

# Critère du «double seuil» de passage

- moyenne aux examens + quiz  $< 50\% \implies$  Échec automatique
- moyenne aux TPs  $< 50\% \implies$  Échec automatique
- Ex : TPs=100% et examens=49%  $\implies$  Échec
- Toutefois, l'inverse n'est pas vrai :
- (examens  $\geq 50\%$  et TPs  $\geq 50\%$ )  $\nRightarrow$  Succès

Référence : voir plan de cours.

# Attribution des notes finales

- Les intervalles pour les notes littérales sont déterminés à la toute fin.
- Les intervalles varient d'une sessions à l'autre.
- La moyenne et l'écart-type n'entrent pas directement dans l'attribution (pas d'attribution avec la courbe normale).
- Toutefois, la difficulté des examens et des travaux pratiques est considérée.
- La moyenne et l'écart-type peuvent servir à évaluer le niveau de difficulté.

Référence : voir plan de cours.

# Entente d'évaluation

Présentation et discussion :

- Échéances et pondérations.
- Discussion.

# Types abstraits de données (TAD)

- Les Types abstraits de données (TAD) sont un sujet central dans INF3105.
- Utilisation transparente au moyen d'une interface publique.
- Abstraction de la représentation interne.
- Principe d'encapsulation.
- Deux visions à développer :
  - Développeur.
  - Utilisateur.

# Le bit comme le TAD de base

- 0 ou 1.
- Signification / Type booléen.
- Interface publique : copie, opérateurs logiques, etc.
- Représentation (courant électrique, onde, mécanique, etc.).
- On peut construire d'autres TAD à partir du bit.
  - Sérialisation.
  - Désérialisation.

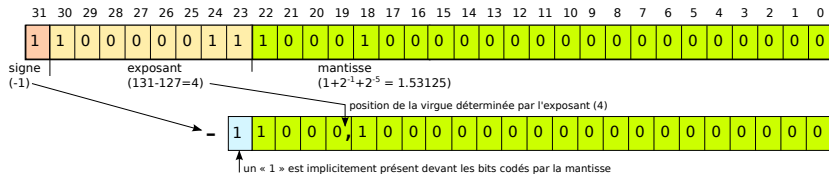


# Les entiers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	0

- Interface publique : copie, opérateurs arithmétiques, comparateurs, etc.
- Représentations :
  - Naïve : compter le nombre de bit à un !
  - En base 2.
  - Complément 2.
- Éviter les bris d'abstraction «dangereux».
  - Exemple : tester le bit de signe pour vérifier si un nombre est négatif : `if (x & 0x80000000)` . Il faut plutôt faire simplement : `if (x<0)`

# Les nombres à virgule flottante



- Interface publique : copie, opérateurs arithmétiques, comparateurs, etc.
- Représentation :
  - Signe
  - Mantisse
  - Exposant

# Caractères

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	—
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

# Génie logiciel

- Branche de l'informatique.
- Processus de développement du logiciel.
- Qualités d'un logiciel.

# Qualités d'un «bon» logiciel

- Qualités :
  - Fonctionnement correct.
  - Robustesse.
  - Efficacité.
  - Maintenabilité.
  - ...
- Certaines qualités en opposition.
- Les TAD et structures de données comme outil.
- Critères de correction des TP.

# Bonne session !