

COORDONNATEUR	LAFORÉST, Louise	laforest.louise@uqam.ca	(514) 987-3000 7790	PK-4725
GROUPE	40 BEAUDRY, Éric	beaudry.eric@uqam.ca	(514) 987-3000 4005	PK-4635

Judi, de 13h30 à 16h30 (cours) – Mardi, de 13h30 à 15h30 (ateliers)

**DESCRIPTION**

Approfondir les connaissances des structures de données et des algorithmes et les appliquer à la résolution de problèmes. Connaître et savoir utiliser des bibliothèques publiques ou normalisées.

Rappels sur les types abstraits de données et sur la complexité des algorithmes. Abstractions de données et de contrôle. Collections et les structures de données nécessaires à leurs réalisations. Arbres, tables, graphes. Utilisation de bibliothèques publiques ou normalisées.

Ce cours comporte une séance obligatoire de laboratoire (2 heures).

Préalables: INF1130 Mathématiques pour informaticien ou MAT2055 Logique et ensembles ; INF2120 Programmation II

**OBJECTIF**

Introduction aux notions de la programmation orientée objet ; utilisation de celles-ci, telles qu'exprimées au moyen du langage C++, pour l'étude et la réalisation de structures de données avancées et des algorithmes associés. Comprendre et maîtriser les structures de données mentionnées dans la description du cours. Atteinte d'un niveau de compréhension permettant de choisir la représentation appropriée pour la solution d'un problème. Être en mesure de comparer l'efficacité des diverses structures d'information pouvant être utilisées dans une application donnée.

ÉVALUATION	Description sommaire	Date	Pondération
	Examen mi-session		25%
	Examen final		25%
	TP 1		15%
	TP 2		20%
	TP 3		15%

L'étudiant doit obtenir une moyenne cumulée aux examens et aux travaux pratiques égale ou supérieure à 50%. Si l'un de ces seuils n'est pas atteint, la mention échec sera automatiquement attribuée au cours et ce, quelles que soient les notes pouvant avoir été obtenues dans l'autre partie. Il n'est donc pas garanti d'obtenir une note supérieure à E même si le seuil aux examens est atteint. La qualité du français est un critère d'évaluation.

À moins d'avis contraire, **les travaux pratiques sont individuels** et en cas de doute sur l'originalité des travaux, un test oral sera exigé.

**En principe, aucun retard n'est toléré** pour les travaux pratiques (voir les normes relatives aux travaux pratiques de votre groupe spécifique); un groupe-cours peut cependant établir une politique particulière permettant un retard avec pénalité. Il est de la responsabilité de l'étudiant de faire des copies de sauvegarde de ses travaux. La perte d'une disquette ou de son contenu n'est pas une raison valable pour remettre un travail en retard ou ne pas le remettre.

**Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants :**

[www.integrite.uqam.ca](http://www.integrite.uqam.ca)

<http://www.bibliotheques.uqam.ca/plagiat>

<http://www.sciences.uqam.ca/decanat/reglements.php>

#### Politique d'absence aux examens

Un étudiant absent à un examen se verra normalement attribuer la note zéro pour cet examen. Cependant, si l'étudiant était dans l'impossibilité de se présenter à l'examen pour un motif valable, certains arrangements pourront être pris avec son enseignant. Pour ce faire, l'étudiant devra présenter à son enseignant l'un des formulaires prévus à cet effet accompagné des pièces justificatives appropriées (par ex., attestation d'un médecin que l'étudiant était dans l'impossibilité de se présenter à l'examen pour des raisons de santé, lettre de la Cour en cas de participation à un jury).

Une absence pour cause de conflit d'horaires d'examen n'est pas considérée comme un motif valable d'absence, à moins d'entente préalable avec la direction du programme et l'enseignant durant la période d'annulation des inscriptions avec remboursement : tel qu'indiqué dans le guide d'inscription des étudiants, il est de la responsabilité d'un étudiant de ne s'inscrire qu'à des cours qui ne sont pas en conflit d'horaire.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique et pour obtenir les

formulaire appropriés, consultez le site web suivant :  
<http://www.info.uqam.ca/enseignement/reglements/politique-dabsence-aux-examens>

## CONTENU

**Introduction et rappels.** Types abstraits de données. Abstraction. Encapsulation. Principes de génie logiciels : qualités d'un logiciel, modularité et généralité.

**Introduction au langage C++.** Fichiers d'entête et sources. Fondements du langage. Mots réservés. Types de base. Variables. Portée des variables. Énoncés et expressions. Opérateurs. Instructions de contrôle d'exécution. Entrées et sorties. Tableaux. Pointeurs et références. Fonctions et types de passage de paramètres. Gestion de la mémoire. Mécanisme de classe. POO en C++. Mécanisme de gabarits (templates) en C++. Gestion d'erreurs et exceptions.

**Algorithmes.** Complexité temporelle et spatiale. Algorithmes de tri. Algorithmes numériques.

**Structures de données linéaires.** Tableaux génériques. Piles. Files. Listes simplement et doublement chaînées. Itérateurs. Applications.

**Structures de données avancées.** Arbres. Définitions. Parcours. Arbres binaires de recherche. Arbres équilibrés. Arbres AVL. Arbres rouge-noir. Itérateurs. Arbres B. Arbres 2-4. Monceaux. Files prioritaires. Arbres spécialisés : arbre d'expressions et codes de Huffman. Applications.

**Adressages dispersé et tables de hachage.** Définitions. Fonction de hachage. Collisions. Gestion des collisions. Applications.

**Bibliothèques normalisées. Standard Template Library (STL).** Exemples d'utilisation.

**Graphes.** Définitions. Représentations. Parcours de graphes : recherche en profondeur; recherche en largeur. Calcul des plus courts chemins : algorithmes de Dijkstra et de Floyd-Washall. Arbre de recouvrement minimal : algorithmes de Kruskal et de Prim-Jarnik. Applications.

## RÉFÉRENCES

- N O Beaudry, É. – *Notes de cours : INF3105 - Structures de données et algorithmes* –
- V R Goodrich, M.T., Tamassia, R., Mount, D. – *Data Structures & Algorithms in C++ – Second edition, Wiley, 2011, 714 pages.*
- V C Stroustrup, B. – *The C++ Programming Language : Special Edition – 3rd edition, Addison-Wesley Professional, 2010, 1040 pages.*
- V C Weiss, M.A. – *Data Structures and Algorithm Analysis in C++ – Addison-Wesley, 2006, 586 pages.*
- V C Gabrini, P. – *Structures de données avancées avec la STL; POO en C++ – Loze-Dion, 2005, 438 pages.*
- V C Delannoy, C. – *Apprendre le C++ – Best of Eyrolles, 2008.*
- V C Josuttis, N. – *The C++ Standard Library - A Tutorial and Reference – Addison-Wesley, 2003, 520 pages.*
- V C Sutter, H., Alexandrescu, A. – *C++ Coding Standards – Addison-Wesley, 2005, 220 pages.*
- V C Vandevorde, D., Josuttis, N. – *C++ Templates - The Complete Guide – Addison-Wesley, 2003, 520 pages.*
- V C Horstmann, C., Budd, T. – *Big C++ – Wiley, 2005, 1186 pages.*
- V C Lippman, S.B., Lajoie, J., Moo, B.E. – *C++ Primer – 4th edition Addison-Wesley 2005, 885 pages.*  
Un livre écrit par des praticiens.
- V C Deitel H. M. et Deitel P. J. – *How to Program – 5th edition Prentice Hall 2003, 1436 pages.*  
(ou la traduction française de la troisième édition du même livre). Très complet sur le langage mais faible sur la STL et les génériques.

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

C : complémentaire – O : obligatoire – R : recommandé