

COORDONNATEUR	BOUKADOUM, Mounir	boukadoum.mounir@uqam.ca	(514) 987-3000 4565	PK-4540
GROUPE	10 DEBAY, Georges	debay.georges@uqam.ca	(514) 987-3000 3699	PK-4115
Lundi, de 18h00 à 21h00 (cours) – Mardi, de 18h00 à 21h00 (ateliers)				

**DESCRIPTION** Introduction à la programmation orienté-objet (OO). Mécanismes d'abstraction et de paramétrisation en OO (dissimulation de l'information, surcharge, généricité, polymorphisme). Étude d'un langage qui supporte l'orienté-objet. Mise en oeuvre de structures de données de base: piles, files, listes chaînées, arbres binaires. Gestion de tables, fouilles séquentielles, logarithmiques. Application aux fichiers. Méthodes de tri. Développement d'applications avec interfaces graphiques. Cours magistral avec séances d'exercices en atelier obligatoires

INF1105 Introduction à la programmation scientifique

- OBJECTIFS**
- Présenter et discuter les structures de données d'usage général: pile, files, arbres et graphes.
  - Apprendre l'utilité des types abstraits de données dans le processus de résolution de problèmes.
  - Implanter les types abstraits de données classiques à l'aide du langage de programmation C++.
  - Transmettre à l'étudiant les connaissances relatives à la programmation par objets ;

ÉVALUATION	Description sommaire	Date	Pondération
	Examen intra		30%
	Examen final		30%
	Travaux pratiques (4)		40%

#### Règle concernant le seuil de passage :

L'étudiant doit obtenir une moyenne cumulée aux examens égale ou supérieure à 50% ainsi qu'une moyenne cumulée pour les travaux pratiques supérieure ou égale à 50%. Si ces seuils ne sont pas atteints, la mention échec sera automatiquement attribuée au cours.

Les règlements de l'UQAM concernant le plagiat seront strictement appliqués.

Les travaux pratiques sont individuels et en cas de doute sur l'originalité des travaux, un test oral pourra être exigé. En cas de plagiat, les sanctions iront de la note zéro à l'exclusion du programme en passant par l'échec au cours.

Une pénalité de 20 points par jour ouvrable de retard sera appliquée aux travaux remis après les dates prévues. Après 5 jours de retard, le travail sera considéré comme non-remis entraînant la note 0.

#### Laboratoires

Les séances d'exercices font partie intégrante du cours. Les étudiants doivent se présenter aux séances pour recevoir les énoncés des exercices.

#### Examens

Aucune documentation permise lors des examens.

#### 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 formulaires appropriés, consultez le site web suivant :

<http://www.info.uqam.ca/enseignement/politiques/absence-examen>

CONTENU	<b>Entrées sorties en C++</b>
	<input type="checkbox"/> Les entrées sorties standards ;

- Les manipulateurs;
- Les fonctions;
- Entrées sorties dans un fichier

**Pointeurs et chaînes**

- Déclaration et initialisation de variables pointeurs;
- Appels de fonctions par référence;
- Tableaux de pointeurs;
- Traitement de caractères et de chaînes de caractères

**Classes et abstraction de données:**

- Types dérivés : pointeurs, vecteurs et structures;
- Allocation dynamique de la mémoire : opérateur new et delete;
- Portée d'une classe et accès aux membres d'une classe;
- Séparation entre l'interface et l'implémentation;
- Contrôle de l'accès aux membres;
- Fonctions : constructeurs, destructeurs, amies, virtuelles; Surcharge des opérateurs;
- Héritage;
- Polymorphisme

**Introduction aux algorithmes**

- Algorithmes de recherche
  - recherche séquentielle;
  - recherche binaire dans un tableau;
  - arbre binaire de recherche
- Algorithmes de tri
  - tri simple
  - tri rapide

**Structures de données élémentaires**

- Notion d'abstraction;
- Types abstraits de données
- Pile
  - définition et exemples;
  - opérations sur les piles;
  - implantation à l'aide des tableaux;
  - modèles de classe;
  - modèles de fonction;
  - implantation de la pile à l'aide de modèles de classe
- File
  - définition et exemples;
  - opérations sur file;
  - implantation à l'aide des tableaux
- Liste
  - définition et exemples;
  - listes simplement chaînées;
  - opérations sur les listes;
  - implantation d'une liste simplement chaînée;
  - implantation à l'aide de tableau;
  - implantation à l'aide des pointeurs;

- implantation de la pile à l'aide de la liste chaînée;
  - comparaison des deux implantations de la file;
  - listes circulaires;
  - opérations de base sur les listes circulaires
- ❑ **Arbres**
- définitions et exemples;
  - opérations de base;
  - algorithmes de parcours d'arbres;
  - arbres binaires;
  - implantation à l'aide des tableaux;
  - implantation à l'aide des pointeurs
- ❑ **Traitement des exceptions**
- Types d'erreurs;
  - Types d'exceptions;
  - Relancement d'une exception;
  - Piège à erreurs

## RÉFÉRENCES

- VO Deitel et Deitel – *Comment programmer en C++, cours et exercices (3ième édition)* – Les éditions Reynald Goulet, 2001.
- VO *Un livre de structure de données en C++*
- VC Michael T. Goodrich, Roberto Tamassia, David Mount – *Data Structures and Algorithms in C++*
- VC Hubbard, J. R. (John Rast) – *Structures de données en C++. Paris – Dunod, 2003.*
- VC Claude Delannoy – *Programmer en langage C++, (4ième édition)* – Eyrolles, 1998.
- VC Delores M. Etter and Jeanine A. Ingber – *Engineering Problem Solving with C++* – Prentice Hall, 2003.
- VC Clifford A. Shaffer – *A practical introduction to data structures and algorithm analysis* – Prentice Hall, 2001.
- VC Larry Nyhoff – *C++: An Introduction to Data Structures, hardcover*
- VC Bruno R. Preiss – *Data structures and algorithms with object-oriented design patterns in C++* – Éditeur: New York: J. Wiley, 1999.
- VC Claude Delannoy – *Programmer en langage C++, (4ième édition)* – Eyrolles, 1998.
- VC Gerhard Willms – *Le Langage C++ (Grand livre)* – Micro Application, 1998. – <http://www.colvir.net/prof/georges.debay/uqam/cours/INF2105/inf2105.htm>
- VC Adam Drozdek – *Data Structures and Algorithms in C++* – PWS Publishing Company, 1996.
- VC Mark Allen Weiss – *Algorithms, data structures, and problem solving with C++* – Addison-Wesley, 1996.
- UC <http://www.colvir.net/prof/georges.debay/uqam/cours/INF2105/inf2105.htm>  
Site du cours

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

C : complémentaire – O : obligatoire – R : recommandé