



DÉPARTEMENT D'INFORMATIQUE

## **420-2C5-LL Programmation Système**

**Pondération**      **2-3-3**  
**Session**            **Automne 2021**

**Dispensé par**      **Yvan Morrissey**  
                         **Local G-204**  
                         [yvan.morrissey@cll.qc.ca](mailto:yvan.morrissey@cll.qc.ca)  
**Responsable**      **Nelson Marceau**

**Adopté par : Département  
d'informatique, mai 2021**

AUTRES MOYENS DE COMMUNICATION AVEC LE PROFESSEUR
En plus des moyens indiqués sur la page couverture du plan de cours, les étudiants peuvent communiquer avec le professeur en tout temps par MIO.

DISPONIBILITÉ ET MODALITÉS DE PARTICIPATION
<p>Les périodes de disponibilité régulières du professeur seront affichées sur le babillard du département, près du local G-204. Pour vous assurer de la disponibilité du professeur pour une rencontre, il est fortement conseillé de prendre rendez-vous au préalable.</p> <p>La présence aux cours et aux périodes d'exercices est fortement recommandée et indispensable pour obtenir un rendez-vous pour des explications individuelles.</p> <p>Chaque semaine, du travail personnel est requis en dehors des heures de cours pour préparer ou finaliser les laboratoires et assimiler la matière vue en classe.</p>

COORDINATION DÉPARTEMENTALE	
Nom	Gilles Champagne
Local	G-204
Courriel	<a href="mailto:gilles.champagne@cll.qc.ca">gilles.champagne@cll.qc.ca</a>

IDENTIFICATION DU COURS	
<b>Titre</b>	Programmation Système
<b>Numéro</b>	420-2C5-LL
<b>Session</b>	Automne 2021
<b>Pondération</b>	2-3-3
<b>Cours préalables</b>	420-1B5-LL Programmation objet
<b>Objectif général</b>	Exploiter les principes de la programmation orientée objet.
COMPÉTENCE(S)	

<b>Numéro</b>	00Q6
<b>Énoncé de la compétence</b>	Exploiter les principes de la programmation orientée objet.
<b>Atteinte</b>	Partielle – Afin que la compétence soit jugée complète, l'étudiant doit réussir les cours : <ul style="list-style-type: none"> <li>• 420-1B5-LL Programmation Objet</li> <li>• 420-2C5-LL Programmation Système</li> </ul>
<b>Éléments de compétence</b>	<ul style="list-style-type: none"> <li>• Analyser le problème</li> <li>• Modéliser les classes.</li> <li>• Programmer des classes.</li> <li>• Produire les algorithmes pour les méthodes.</li> <li>• Générer l'interface graphique.</li> <li>• Documenter la programmation.</li> <li>• Appliquer la procédure liée à la gestion des versions de programmes.</li> </ul>

#### PLACE DU COURS DANS LE PROGRAMME

##### Session dans le programme : Session 3

- Ce cours requiert les connaissances du cours 420-1B5-LL Programmation Objet.
- Ce cours est le troisième d'une série prévue en programmation 420-2C5-LL Programmation Système.

Suite de cours en développement Web prévue au programme :

##### Tronc commun

Session 3

420-3A5-LL Interfaces Web

Session 3

420-4C5-LL Base de données et sécurité des applications

##### Profil Développement d'applications

Session 4

420-2D7-LL Développement Web

STRATÉGIES D'ENSEIGNEMENT ET D'APPRENTISSAGE							
<b>Méthodologie</b>	2 heures de théorie en présentielles et 3 heures de laboratoire en présentielles obligatoires sont prévues à chaque semaine. 2 heures de travail personnel sont aussi prévues pour permettre à l'étudiant d'effectuer des lectures et des recherches ainsi que compléter les travaux.						
<b>Travail attendu des étudiants</b>	De plus, l'étudiant a la responsabilité de se présenter aux périodes prévues pour les évaluations. Le droit de reprise d'une évaluation pour un étudiant absent ne s'exerce pas automatiquement. L'étudiant a la responsabilité de prévenir le professeur de son absence et de discuter avec lui des modalités de reprise.						
<b>Évaluations</b>	<table> <tr> <td>3 Travaux pratiques</td><td>15 % (5% pour chacun)</td></tr> <tr> <td>1 projet Git</td><td>25 %</td></tr> <tr> <td>3 Examens</td><td>60 % (20% pour chacun)</td></tr> </table>	3 Travaux pratiques	15 % (5% pour chacun)	1 projet Git	25 %	3 Examens	60 % (20% pour chacun)
3 Travaux pratiques	15 % (5% pour chacun)						
1 projet Git	25 %						
3 Examens	60 % (20% pour chacun)						
<b>Seuil de réussite</b>	Une moyenne cumulative de 60% ou plus doit être atteinte pour réussir le cours.						

CONTENU DE COURS		
Semaine	Théorie	Laboratoire
<b>Semaine 1</b>	<b>Plan de cours</b> <b>Git vs GitHub</b> <ul style="list-style-type: none"> <li>À propos de la gestion de version</li> <li>Développements distribués</li> <li>Contribution à un projet</li> <li>Maintenance d'un projet</li> </ul>	<b>Laboratoire #1</b> Conversion <b>(Formatif)</b> Projet Visual Studio 2019 avec Git
<b>Semaine 2</b>	<b>Masque et décalage</b> <b>Symbole Pour le Masque</b> <ol style="list-style-type: none"> <li>&amp; et logique</li> <li>  ou Logique</li> </ol> <b>Symbole pour le décalage</b> <ol style="list-style-type: none"> <li>&gt;&gt; décalage vers la droite, division</li> <li>&lt;&lt; décalage vers la gauche multiplication</li> </ol>	<b>Tp#1 (5%)</b>
<b>Semaine 3</b> <b>Semaine 4</b>	<b>Les Pointeurs</b> <ul style="list-style-type: none"> <li>Qu'est-ce qu'un Pointeur ?</li> <li>Déclaration</li> <li>L'Opération &amp;, *</li> <li>Comparaison avec une variable classique</li> </ul>	<b>Laboratoire #2 (Formatif)</b>

Semaine 5	<b>Examen #1 (20%)</b>	
Semaine 6 Semaine 7	<b>Programmation gestion processus</b> Qu'est-ce qu'un thread ? <ul style="list-style-type: none"> <li>• Les threads</li> <li>• La synchronisation</li> <li>• Sémaphore</li> <li>• Mutex</li> </ul>	<b>Tp#2 (5%)</b>
Semaine 8 Semaine 9	<b>Programmation gestion de processus</b> La communication interprocessus ( <b>Les sockets</b> ) <ul style="list-style-type: none"> <li>• Qu'est-ce qu'un Socket</li> <li>• Utilisation du Socket</li> <li>• Commandes reliées au sockets</li> <li>• Modèle client-serveur</li> <li>• Adresses</li> </ul>	<b>Tp#3 (5%)</b>
Semaine 10	<b>Examen #2 (20%)</b>	
Semaine 11	<b>Documenter la programmation.</b> <ul style="list-style-type: none"> <li>• Notation claire de commentaires dans le code informatique.</li> <li>• Notation claire de la documentation d'aide à la programmation.</li> <li>• Utilisation appropriée des générateurs de documentation.</li> </ul> Ex : <b>Doxygen</b> <b>Générer l'interface graphique.</b> <ul style="list-style-type: none"> <li>• Choix approprié des éléments graphiques pour l'affichage et la saisie.</li> <li>• Positionnement correct des éléments graphiques.</li> <li>• Paramétrage correct des éléments graphiques.</li> </ul>	<b>Projet Client-Serveur (25%)</b>
Semaine 12 Semaine 13	<b>Les branches avec Git</b> <ul style="list-style-type: none"> <li>• Ce qu'est une branche</li> </ul> Gestion de branches <b>Git distribué</b> <ul style="list-style-type: none"> <li>• Créer une branche</li> <li>• Changer de branche</li> <li>• Faire un "commit"</li> <li>• Merge votre branche vers une autre</li> <li>• Push une branche vers un master</li> <li>• Supprimer une branche</li> </ul>	<b>Projet Client-Serveur (Suite)</b>
Semaine 14	<b>Utilitaires Git</b> <ul style="list-style-type: none"> <li>• Sélection des versions</li> <li>• Indexation interactive</li> <li>• Le remisage</li> <li>• Réécrire l'historique</li> </ul>	
Semaine 15	<b>Examen #3 (20%)</b>	

TECHNOLOGIES VUES DANS LE COURS
Visual Studio, GIT

MATÉRIEL DE COURS
- Notes de cours Disponible sur LEA/Document

MÉDIAGRAPHIE	
Ouvrages	Notes de cours

Mesures et protection individuelle	
Covid-19	<ol style="list-style-type: none"><li>1- Dans tous les déplacements, masque lavable ou jetable en tout temps</li><li>2- Dans la classe théorique où les déplacements sont quasi nuls et que la distance de 1.5m peut être respectée, le couvre-visage peut être enlevé</li><li>3- Dans les lieux où les déplacements sont fréquents et où on ne peut respecter le 2m, port du masque en tout temps</li><li>4- Pour des échanges où on ne peut respecter le 1.5m de distance pour un cumul de plus de 15 minutes dans la journée (questions en laboratoire, par exemple), port du masque jetable et de la visière (ou lunettes de protection). Jeter le masque à la fin de la journée.</li><li>5- En entrant et en sortant de la classe, se laver les mains.</li><li>6- Lors de l'utilisation de postes informatiques, se laver les mains et désinfecter le matériel avant et après utilisation.</li><li>7- Pour toute manipulation de matériel, se laver les mains avant et après.</li></ol>

POLITIQUE D'ÉVALUATION DES APPRENTISSAGES	
<b>Annonce des évaluations</b>	Selon l'article 1.7 de la PIEA, « (...) le professeur présente, au moins une semaine avant une évaluation sommative, ses attentes, ses exigences, les critères d'évaluation, le matériel autorisé, la durée, la pondération et les critères d'évaluation des

<p><b>Évaluations et révision de notes</b></p>	<p>objectifs langagiers. Il précise également les éléments du contenu du cours sur lesquels l'évaluation portera. »</p> <p>Selon l'article 1.10 de la PIEA, « Le professeur communique aux étudiants, au plus tard quinze (15) jours ouvrables suivant la passation d'une situation d'évaluation à l'enseignement régulier, les résultats qu'ils ont obtenus à ces évaluations (cette disposition ne s'applique pas à la note finale). »</p> <p>Selon l'article 1.11 de la PIEA, « Le professeur permet à tout étudiant de consulter la copie corrigée de son examen ou de son travail. »</p> <p>Selon l'article 10.1.1 de la PIEA, « L'étudiant a la responsabilité de consulter la copie corrigée de son évaluation sommative et de rassembler tous les éléments en sa possession sur lesquels il fonde sa demande de révision de notes. »</p> <p>Selon la section 10.2 de la PIEA :</p> <ul style="list-style-type: none"> <li>• « Pour tout résultat communiqué en cours de session, l'étudiant qui veut voir un résultat révisé doit, dans un délai de cinq (5) jours ouvrables suivant la communication de la note (...) ».</li> <li>• « Pour tout résultat communiqué après la session, l'étudiant doit soumettre sa demande de révision de notes le plus tôt possible, mais au plus tard cinq (5) jours ouvrables après la date officielle de dépôt des notes de la session.(...) ».</li> </ul> <p>** Pour connaître toutes les règles et modalités de demande de révision de note, consulter la section 10 de la PIEA, en lien ci-bas.</p>
<p><b>Lien vers la PIEA</b></p>	<p>Politique institutionnelle d'évaluation des apprentissages :  <a href="http://cll.qc.ca/admin/wp-content/uploads/2018/06/2222-00-10-Pol-institutionnelle-évaluation-apprentissages-2018-06-11.pdf">http://cll.qc.ca/admin/wp-content/uploads/2018/06/2222-00-10-Pol-institutionnelle-évaluation-apprentissages-2018-06-11.pdf</a></p>
<p><b>Lien vers les RDEA</b></p>	<p>Règles départementales d'évaluation des apprentissages :</p>
<p><b>Lien vers la PVL</b></p>	<p>Politique de la valorisation de la langue :  <a href="http://cll.qc.ca/admin/wp-content/uploads/2012/05/2231-00-10.pdf">http://cll.qc.ca/admin/wp-content/uploads/2012/05/2231-00-10.pdf</a></p>

	<p><b><u>Description Programmation Système</u></b></p> <p>La <b>programmation système</b> est un type de <a href="#">programmation</a> qui vise au développement de <a href="#">programmes</a> qui font partie du <a href="#">système d'exploitation</a> d'un <a href="#">ordinateur</a> ou qui en réalisent les fonctions. Elle se distingue de la <a href="#">programmation des applications</a> en ce qu'elle s'intéresse non pas au <a href="#">traitement des données</a>, mais à la résolution des problèmes pour les humains, aux <a href="#">interfaces</a>, aux <a href="#">protocoles</a> et à la gestion des ressources, telles que le temps et l'espace. Donc, en réalité seuls les programmes d'application sont réellement utilisés par les utilisateurs. Les programmes système le sont implicitement.</p> <p>Elle inclut, en outre, l'accès aux <a href="#">fichiers</a>, la programmation du <a href="#">clavier</a>, de l'<a href="#">écran</a>, des <a href="#">modems</a>, la programmation <a href="#">réseau</a>, et, en général, la programmation de tous les <a href="#">périphériques</a> qui font entrer ou sortir de l'information d'un ordinateur, de la <a href="#">mémoire vive</a> et des <a href="#">processeurs</a>.</p> <p><b><u>Langage</u></b></p> <p>La programmation système se fait généralement par le biais de langages tel que le langage <a href="#">assembleur</a> et d'un langage de bas niveau (<a href="#">langage C</a>), c'est le cas des <a href="#">systèmes d'exploitation</a> de type <a href="#">UNIX</a> (<a href="#">GNU/Linux</a>, <a href="#">FreeBSD</a>, <a href="#">Solaris</a>...) dont 90 % du code est écrit en <a href="#">C</a>, le reste (10 %) est en assembleur suivant les architectures cibles (<a href="#">x86</a>, <a href="#">SPARC</a>...).</p> <p><a href="https://fr.wikipedia.org/wiki/Programmation_syst%C3%A8me">https://fr.wikipedia.org/wiki/Programmation_syst%C3%A8me</a></p>
--	---