

PLAN DE COURS

Programmation distribuée
420-3N5-RA
420.B0-Techniques de l'informatique
2-3-3

Automne 2024
Frédéric Bergeron
Département des techniques de l'informatique
4044-D
418-338-8591 Poste 128
fbergeron@cegepthetford.ca

INFORMATIONS SUR LE DÉROULEMENT DU COURS

Plages horaires du cours	Jour	Heures	Commentaires
	Mardi	14h55 – 17h35	Aucun
	Jeudi	13h05 – 15h45	Aucun
Cours théoriques et laboratoires	Tous les cours théoriques et laboratoires se dérouleront en présence au cégep.		
Disponibilités de l'enseignant	Rendez-vous disponibles sur demande via MIO ou Teams. Les messages seront répondus dans un délai de 24h ouvrable. L'enseignant est disponible durant les périodes indiquées par la couleur verte sur la porte de son bureau.		
Plateformes utilisées	<input checked="" type="checkbox"/> Moodle <input checked="" type="checkbox"/> Léa <input checked="" type="checkbox"/> Gitlab/GitHub		Les notes de cours et les énoncés seront disponibles sur Moodle. Les résultats d'évaluation seront publiés sur Léa. Les évaluations seront à remettre en main propre, sur Moodle ou sur GitHub, tel qu'indiqué dans les énoncés.

Place dans le programme

Ce cours se situe à la cinquième session du programme de formation. L'étudiant a acquis différentes compétences de programmation, de réseau et de Web. Il a donc atteint le niveau requis pour s'attaquer aux applications traitant les échanges de données. Le présent cours lui permettra d'effectuer le développement de services d'échange de données. Ces services peuvent alimenter des applications natives, Web ou des objets connectés. Le développement de services d'échanges de données peut se faire pour de nouvelles applications ou des applications à modifier. Il s'effectue à partir de documents de conception et à l'aide de procédures de suivi des problèmes et de gestion des versions.

Apport à la formation

Ce cours prend son origine d'une compétence au choix du devis ministériel du programme de techniques de l'informatique (420.B0). Cette compétence 00SV concerne le développement de services d'échanges de données. Ces services peuvent alimenter des applications natives, Web ou des objets connectés. Ce cours consolide diverses notions couvertes dans les sessions précédentes comme la programmation, les bases de données et la sécurité informatique.

Le préalable de ce cours est le cours « Programmation orientée objet avancée ». Il est préalable à deux cours « Intégration Web » et « Projet de fin d'études ».

<i>Objectif</i>	<i>Standard</i>
Énoncé de la compétence	
Effectuer le développement de services d'échange de données <div style="text-align: right;">Code : 00SV</div>	
Contexte de réalisation	
<ul style="list-style-type: none"> - Pour des services d'échange de données alimentant des applications natives, Web ou des objets connectés. - Pour de nouvelles applications et des applications à modifier. - À partir des documents de conception. - À l'aide de procédures de suivi des problèmes et de gestion des versions. 	
Éléments de la compétence	Critères de performance
1. Analyser le projet de développement de l'application.	<ul style="list-style-type: none"> • Analyse juste des documents de conception. • Détermination correcte des tâches à effectuer.
2. Préparer l'environnement de développement informatique.	<ul style="list-style-type: none"> • Installation correcte de la plateforme de développement et du système de gestion de base de données de développement. • Installation correcte des logiciels et des bibliothèques. • Configuration appropriée du système de gestion de versions. • Importation correcte du code source.
3. Préparer la base de données.	<ul style="list-style-type: none"> • Création ou adaptation correctes de la base de données. • Insertion correcte des données initiales ou des données de tests. • Respect du modèle de données.
4. Programmer la logique applicative du service.	<ul style="list-style-type: none"> • Programmation ou intégration correctes de mécanismes d'authentification, d'autorisation ou d'établissement de liaison sécurisée. • Programmation correcte de la réception des données d'entrée. • Choix approprié des clauses, des opérateurs, des commandes ou des paramètres dans les requêtes à la base de données. • Manipulation correcte des données de la base de données. • Programmation correcte de l'envoi des données de sortie.

	<ul style="list-style-type: none"> • Application rigoureuse des techniques de programmation sécurisée. • Respect des protocoles de communication et des formats d'échange de données.
5. Programmer une application de test en utilisant le service.	<ul style="list-style-type: none"> • Récupération exacte de l'interface du service. • Utilisation appropriée du service. • Conversion appropriée des données fournies par le service en données exploitables par l'application de test.
6. Contrôler la qualité du service.	<ul style="list-style-type: none"> • Application rigoureuse des plans de tests. • Revues de code et de sécurité rigoureuses. • Pertinence des correctifs. • Respect des procédures de suivi des problèmes et de gestion des versions. • Respect des documents de conception.
7. Participer au déploiement du service.	<ul style="list-style-type: none"> • Application correcte de la procédure de migration du service sur le serveur. • Application rigoureuse des mesures de sécurité.
8. Rédiger la documentation.	<ul style="list-style-type: none"> • Détermination correcte de l'information à rédiger. • Notation claire du travail effectué.

HABILETÉS GÉNÉRIQUES

<u>Attitudes</u>	<u>Habilités intellectuelles</u>	<u>Techniques d'apprentissage</u>	<u>Langage</u>
<ul style="list-style-type: none"> • Esprit critique • Soucis de la rigueur • Capacité d'adaptation 	<ul style="list-style-type: none"> • Capacité d'abstraction • Reconnaître et appliquer des modèles • Identifier les éléments d'une situation • Exécuter une démarche de résolution 	<ul style="list-style-type: none"> • Gestion de temps • Techniques d'acquisition de connaissances • Démarche de résolution de problèmes • Prise de notes et utilisation des notes prises 	<ul style="list-style-type: none"> • Capacité d'expression écrite en français • Lecture/compréhension minimale d'éléments techniques en langue anglaise • Termes techniques à maîtriser

PARTIE 1 : Approche client-serveur.

Objectifs d'apprentissage de la partie

- À la fin de cette partie, l'étudiant sera en mesure de comprendre les mécanismes de communication entre un client et un serveur et les protocoles de la couche réseau mis en jeu. Il sera en mesure de développer des applications natives, côté client et côté serveur, selon des protocoles établis, en tenant compte des exigences de traitements concourants.

Déroulement

	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'ENSEIGNEMENT ET D'APPRENTISSAGE
Cours 1 à 9 Nombre de périodes : 23	<ul style="list-style-type: none">• Préparer l'environnement de développement.• Rappel de quelques notions de réseau.• Notions de télématique.• Communication orientée connexion (TCP) et communication non orientée connexion (UDP).• Structuration des échanges et protocoles.• Gestion du parallélisme.• Survol des problèmes classiques en parallélisme.• Notion de sockets.	<ul style="list-style-type: none">• Présentation théorique des éléments reliés.• Questionnement d'intégration.• Cas d'étude.• Modélisation de protocole.• Observation d'échanges entre applications.• Exemples typiques et exercices dirigés.
	GESTION DE L'ÉTUDE — TRAVAIL HORS CLASSE	
	<ul style="list-style-type: none">• Utilisation de la documentation officielle pour consolider les notions vues.• Réalisation de divers exercices et exemples fournis par l'enseignant.	
	ÉVALUATIONS	
	<ul style="list-style-type: none">• Exercices notés (Contribuants au 25% associé)• Examens 1 (15%)	

PARTIE 2 : Création et consommation de services.

Objectifs d'apprentissage de la partie

- À la fin de cette partie, l'étudiant sera en mesure de créer des API, de les sécuriser et de les consommer. Des documents de conception serviront à développer diverses applications. Différents langages, infrastructures et bases de données seront agencés dans une infrastructure distribuée.

Déroulement

	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'ENSEIGNEMENT ET D'APPRENTISSAGE
Cours 10 à 18 Nombre de périodes : 23	<ul style="list-style-type: none">• Création de la structure d'un projet.• Implémentation adéquate.• Bonne gestion de codes d'utilisation.• Création et/ou adaptation correctes de la base de données.• Manipulation correcte des données de la base de données.• Insertion correcte des données initiales ou des données de tests.• Test de l'API en utilisant des outils divers.• Migration du service sur le serveur.• Application de mesures de sécurité (authentification et autorisation).• Création d'applications de test pour consommer ces API.• Gestion de version.• Architecture en microservices.• Services composés.• Serveurs mandataires.• Services d'abonnement et de publication• Réplication de serveurs.	<ul style="list-style-type: none">• Présentation théorique des éléments reliés.• Questionnement d'intégration.• Preuves de concept, cas d'étude et exemples typiques.• Démonstrations.
	GESTION DE L'ÉTUDE — TRAVAIL HORS CLASSE	
	<ul style="list-style-type: none">• Utilisation de la documentation officielle pour consolider les notions vues.• Réaliser divers exercices et exemples fournis par l'enseignant.	
	ÉVALUATIONS	
	<ul style="list-style-type: none">• Exercices notés (Contribuants au 25% associé)• Examens 2 (20%)	

PARTIE 3 : Intégration de services externes

Objectifs d'apprentissage de la partie

- À la fin de cette partie, l'étudiant sera en mesure de consommer différents services externes par diverses applications selon des documents de conception en utilisant les plus récentes technologies. Il sera en mesure d'intégrer plusieurs services au sein d'une application en la configurant et en la programmant efficacement.

Déroulement

	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'ENSEIGNEMENT ET D'APPRENTISSAGE
Cours 19 à 24 Nombre de périodes : 15	<ul style="list-style-type: none">• Préparer l'environnement de développement.• Migration du service sur le serveur.• Application de mesures de sécurité (authentification et autorisation).• Utilisation d'objets domotiques.• Utilisation et implémentation d'une plateforme de collecte de données.• Création d'applications de test pour consommer le service.• Gestion de version.• Présentation des données.	<ul style="list-style-type: none">• Présentation théorique des éléments reliés.• Questionnement d'intégration.• Mises en situation.• Preuves de concept, cas d'étude et exemples typiques.• Démonstrations.
	GESTION DE L'ÉTUDE — TRAVAIL HORS CLASSE	
	<ul style="list-style-type: none">• Exercices fournis par l'enseignant.• Exemples de la documentation officielle.	
	ÉVALUATIONS	
	<ul style="list-style-type: none">• Exercices notés (Contribuants au 25% associé)	

PARTIE 4 : Projet d'applications dans des infrastructures en temps réel.

Objectifs d'apprentissage de la partie

- À la fin de cette partie, l'étudiant sera en mesure de comprendre les mécanismes d'échange en temps réel, avec les contraintes associées. Il sera également en mesure d'exploiter efficacement les plateformes supportant le temps réel. Par exemple l'utilisation des queues de message dans des applications dans un contexte d'usine 4.0.

Déroulement

	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'ENSEIGNEMENT ET D'APPRENTISSAGE
Cours 25 à 30 Nombre de périodes : 15	<ul style="list-style-type: none">• Préparer l'environnement de développement.• Durée de vie des messages• Modes de connexion et échanges• Plateformes supportant le temps réel• Couplage• Présentation des données.	<ul style="list-style-type: none">• Présentation théorique des éléments reliés.• Mises en situation spécifiques• Présentation d'exemples et cas d'étude variés pour montrer l'utilité et la pertinence de ces mécanismes.
	GESTION DE L'ÉTUDE — TRAVAIL HORS CLASSE	
	<ul style="list-style-type: none">• Exemples de la documentation officielle.• Réalisation du travail synthèse.	
	ÉVALUATIONS	
	<ul style="list-style-type: none">• Travail synthèse (15%)	

SOMMAIRE DES ÉVALUATIONS

Moment	Évaluation	Pondération
Tout au long de la session	Exercices notés	Pondération égale 25%
Cours # 9	Examen 1	15%
Cours # 18	Examen 2	20%
Cours # 26	Examen synthèse	20%
Cours # 30	Remise du travail synthèse	20%

Modalités particulières d'évaluation

Les travaux réalisés dans le cadre de ce cours seront soumis aux contraintes suivantes :

- 60% des points sont amassés durant le travail en classe
- 40% des points sont amassés durant le travail hors-classe

Les balises entourant l'utilisation de l'intelligence artificielle générative sont précisées dans le RDÉA. Des balises supplémentaires pourront être précisées dans les consignes des travaux.

DESCRIPTION DE L'ÉVALUATION FINALE

Description de l'évaluation	Critères d'évaluation	
<p>L'évaluation finale est formée par l'examen synthèse (20%) et le travail synthèse (TS) (20%).</p> <p>L'évaluation finale porte sur les 4 parties du cours.</p> <p>Le travail pratique couvre les aspects pratiques alors que l'examen synthèse couvre les aspects théoriques.</p> <p>Cours : moment où aura lieu l'évaluation finale (temps de travail en classe et hors-classe pour le TS et rencontre 26 pour l'examen synthèse).</p>	Implémentations correctes et complètes.	
	Réponses justes, concises et précises.	
	Intégration juste de divers protocoles d'échange de données.	
	Programmation correcte.	
	Fonctionnement des applications.	
	PONDÉRATION	40 %