



CÉGEP DE SAINT-HYACINTHE
3000, rue Boullé, Saint-Hyacinthe
Québec J2S 1H9
Téléphone : (450) 773-6800
Montréal : (514) 875-4445

Développement d'applications distribuées
420-GEN-HY
(3-3-3)

PLAN DE COURS

Session Automne 2016

Nom de l'enseignant: Mustapha Boushaba
Bureau : B-2344c
Poste téléphonique : 2564
Adresse électronique: par MIO

1. DESCRIPTION

Le cours aborde le vaste monde des applications distribuées. Il présente différentes architectures et approches de programmation réseau, traitement parallèle, informatique distribuée, gestion de concurrence et synchronisation des processus et des données.

2. COMPÉTENCE ET ÉLÉMENTS DE COMPÉTENCE VISÉS

0174 Mettre à profit les possibilités d'un environnement informatique en réseau

1. Utiliser un système d'exploitation de réseau.
2. Relier une station de travail au réseau.
3. Analyser l'architecture d'un réseau.
4. Assurer l'échange des données entre des stations de travail compatibles reliées en réseau.

3. OBJECTIFS INTÉGRATEURS

Au terme de ce cours d'exploration, l'étudiant sera en mesure de comprendre le fonctionnement des différents types d'applications distribuées et leur efficacité respective selon les contextes. Il aura acquis une base solide pour commencer à exploiter efficacement les possibilités d'un environnement informatique en réseau.

4. PLACE DANS LE PROGRAMME

Il est essentiel d'avoir réussi le cours Architecture des applications 420-GEK-HY pour être en mesure d'aborder les notions d'applications distribuées.

5. OBJECTIFS D'APPRENTISSAGE

Dans le cadre de ce cours, nous utiliserons plusieurs logiciels qui nécessitent le Framework .NET.

Les objectifs du cours sont:

- 1- Utiliser un système d'exploitation de réseau.
- 2- Relier une station de travail au réseau.
- 3- 3. Analyser l'architecture d'un réseau.

- 4- Assurer l'échange des données entre des stations de travail compatibles reliées en réseau.

6. Critères de performance

1

- 1.1 Examen des caractéristiques et des fonctions des éléments physiques du réseau.
- 1.2 Examen des caractéristiques et des fonctions des éléments logiques du réseau.
- 1.3 Examen des caractéristiques de l'organisation des données dans le réseau.
- 1.4 Création correcte des comptes et des groupes d'utilisatrices et d'utilisateurs du réseau.
- 1.5 Mise en place des mesures de sécurité et de protection des comptes et des groupes.
- 1.6 Attribution des droits et des restrictions d'accès aux ressources du réseau.
- 1.7 Automatisation correcte des tâches liées à l'utilisation d'une application.

2

- 2.1 Installation correcte des éléments physiques de la station de travail.
- 2.2 Détermination appropriée des paramètres de configuration de la station.
- 2.3 Installation correcte des logiciels d'accès au réseau.
- 2.4 Vérification du fonctionnement de la station dans l'environnement réseau.

3

- 3.1 Examen de la topologie d'un réseau.
- 3.2 Examen des modes et des protocoles de transport des données associés aux différents supports de transmission.
- 3.3 Examen des protocoles de transfert de l'information entre les différents nœuds d'un réseau.
- 3.4 Examen des protocoles de communication entre les stations d'un réseau.
- 3.5 Examen des protocoles de présentation, de compression et de protection des données.
- 3.6 Examen des protocoles liés aux différentes classes d'applications distribuées.

4

- 4.1 Choix du protocole de communication approprié.
- 4.2 Choix de la méthode d'interaction appropriée à la nature de l'application distribuée.
- 4.3 Utilisation judicieuse des fonctions de programmation associées aux protocoles.
- 4.4 Création des programmes de soutien appropriés aux protocoles retenus.
- 4.5 Vérification du fonctionnement des programmes.

7. Savoirs essentiels

- Modèle OSI
- Sessions, authentification et sécurisation des communications
- Modèles SOA/EAI
- Partage de données par passage de messages
- Gestion explicite des contraintes de temps

- Gestion des nœuds et processus
- Configurer une application distribuée
- Connaître les différentes perspectives d'architecture :
- Connaître les protocoles de services (SOAP, JSON, ...)
- Comprendre et savoir palier au point individuel de défaillance (SPOF)
- Comprendre les mécanismes de routages d'applications distribuées
- Comprendre les stratégies de scalabilité des principales architectures d'application distribuées.
- Savoir sélectionner une architecture d'application distribuée
- Connaître les stratégies de suivi et de relève applicative des principales architectures d'applications distribuées.

8. ÉTAPES

- 1- Rappel sur les systèmes informatique et les mégadonnées (données massives) (3h)**
- 2- Paradigme des langages de programmation (3h)**
- 3- Les systèmes distribués (6h)**
 - Intérêt des systèmes distribués
 - Conception
 - Architecture
 - Les réseaux de télécommunication et le modèle en couche OSI
 - Place aux services web
- 4- Programmation distribuées (3h)**
 - Principes de la programmation distribuée
 - Les langages de la programmation distribuée
- 5- Programmation fonctionnelle (3h)**
 - Exploration du paradigme de la programmation fonctionnelle
- 6- Introduction et présentation du langage F# (30h)**
 - Les types de données et les expressions
 - Les objets mutables et immutables
 - Expressions conditionnelles et définitions
 - Les fonctions
 - Fonctions anonymes, application partielle et composition
 - Fonctions récursives
 - Fonctions d'ordre supérieur
 - Pattern matching
 - L'expression objet
- 7- Programmation parallèle (6h)**
 - paradigme Map/Reduce

8- Introduction à la programmation concurrente (3h)

- Types de concurrence
- Avantages de la concurrence
- Modélisation de la concurrence
- Les langages de programmation concurrente et le traitement parallèle

9- Introduction à la programmation orienté processus (15h)

- Les processus : concept, opérations et relations
- Fils d'exécution (threads)
- La synchronisation
- La communication entre processus (mémoire commune, passage de messages)
- Bases d'Erlang (types, les conditions, les fonctions, pattern matching, etc.)
- Les fonctions et modules
- Création et mise à jour des processus

10- Introduction à l'infonuagique (9h)

- Présentation du concept
- Caractéristiques
- Quelques scénarios
- Les différentes formes de l'infonuagique (IaaS, PaaS, SaaS, privé, public, hybride)
- Modèles de service
- Cycle de vie d'un service
- Modèles de déploiement
- Déploiement d'un service

9. ACTIVITÉS D'ENSEIGNEMENT ET D'APPRENTISSAGE

Compte tenu de la nature transversale de ces savoirs essentiels, ces derniers seront étudiés dans chaque paradigme présenté dans les principaux thèmes du cours décrit plus haut.

Les architectures peuvent être présentées sous forme de démonstrations, présentations ou de projets alors que les notions de programmation fonctionnelle sont d'excellents candidats pour les travaux pratiques.

Des cours magistraux, des études de cas, des exercices avec des sessions de pratique sur ordinateur constitueront l'une des approches retenues pour l'apprentissage. Vous serez appelés à suivre des tutoriels sur Internet.

Des exercices dirigés et des présentations en classe avec les logiciels ainsi que des travaux pratiques viendront faciliter l'assimilation des nouvelles connaissances.

L'étudiant devra appliquer concrètement les notions vues en classe, faire appel à son imagination, consigner des commentaires pertinents et déboguer les applications.

10.ÉVALUATION SOMMATIVE / CALENDRIER DES ÉVALUATIONS

Examen de mi- session : 20%
(TP1+Examen final) : 55%

Épreuve terminale de cours

	Contexte	Matière	Thème	Durée	Date d'énoncé	Pondération
TP 1	Individuel	1-4	Recherche sur les services web + implémentation + présentation	3 semaines	Semaine 1	20%
TP2	Équipe de 2	1- 6	F#	3 semaines	Semaine 6	15%
TP3	Équipe de 2	1-9	Erlang	2 semaines	Semaine 9	10%
Examen Mi-session	Individuel	1-6		2h30	Semaine 7	20%
Examen final	Individuel	1-10		2h30	Semaine 15	35%

Les connaissances en informatique sont cumulatives. La matière susceptible d'être utilisée à l'examen porte donc sur l'ensemble du cours.

Les dates sont à titre indicatif seulement et peuvent varier en fonctions de plusieurs facteurs, une date de remise plus précise vous sera communiquée avec chaque travail.

11.QUALITÉ DU FRANÇAIS

10% de la note des travaux et examens est attribué à la qualité du français. Prenez donc soin de vos présentations.

12.RÈGLES PARTICULIÈRES D'ÉVALUATION DES APPRENTISSAGES

SVP vous référez au « GUIDE DE PRÉSENTATION D'UN TRAVAIL ÉCRIT » disponible sur le portail pour la présentation des travaux écrits à remettre.

Les règles de la PIEA seront appliquées intégralement, et un minimum de 60% sera exigé pour les examens et de 60% aussi pour les travaux.

13.RÈGLES IMPORTANTES

Toutes les règles de la Politique Institutionnelle d'Évaluation des Apprentissages (PIEA) s'appliquent. S'ajoutent les règles suivantes :

- Il est rigoureusement interdit de boire ou manger dans les laboratoires informatiques.
- La personne absente a la responsabilité de rattraper auprès de ses collègues la matière manquée.
- Toute absence devra être justifiée dans les 48h auprès du secrétariat et le « papier rose» devra être remis au professeur.
- Une absence non motivée à une évaluation équivaut à un 0. Une absence motivée permet un report de l'évaluation après entente avec le professeur.
- Aucun écart de langage n'est toléré, que ce soit entre étudiants ou avec le professeur. Il en résulte l'exclusion pour le reste de la période, sans autre préavis.
- Les cellulaires et tablettes sont formellement interdits sauf mention spéciale du professeur pour application au cours.
- Les cas de triche et de plagiat sont sanctionnés par un 0 et une mention au dossier.
- Les travaux devront être remis aux dates prévues par le professeur. Aucun retard ne sera toléré.

14.ÉVALUATION FORMATIVE

Les laboratoires de la session correspondent aux évaluations formatives de ce cours.

15.ÉPREUVE TERMINALE DU COURS

Compte-tenu de la nature exploratoire de ce cours, l'épreuve terminale comportera une importante portion théorique sur le fonctionnement des applications distribuées. Des études de cas pourront être utilisées pour valider la compréhension des différents contextes où ces technologies seraient utilisées adéquatement. La programmation fonctionnelle pourrait faire l'objet d'un examen pratique. Les exercices soumis durant les cours correspondent aux évaluations formatives de ce cours. Les examens feront objet d'épreuve terminale de cours.

16.MATÉRIEL REQUIS

Vous aurez besoin d'un ordinateur PC doté d'un système d'exploitation (Windows 7 ou plus) avec une connexion Internet. D'autres outils vous seront demandés pour installation. Entre autres, F# et Erlang.

17. MÉDIAGRAPHIE

Aucun livre de cours n'est exigé. Cependant, les références suivantes peuvent s'avérer utiles :

- **Seven Concurrency Models in Seven Weeks.**

Paul Butcher, Éditions The Pragmatic Bookshelf 2014

ISBN 978-1-93778-565-9

<https://pragprog.com/book/pb7con/seven-concurrency-models-in-seven-weeks>

- **Functional Thinking**

Neal Ford, Éditions O'Reilly 2014

ISBN 978-1-449-36551-6

<http://shop.oreilly.com/product/0636920029687.do>

- **Beginning F# 4.0**

Par Robert Pickering, Kit Eason

ISBN 978-1-4842-1374-2

<http://www.springer.com/us/book/9781484213759>

- **Programmer en Erlang**

Francesco Cesarini, Simon Thompson

ISBN 9782744024450

<http://www.archambault.ca/cesarini-francesco-programmer-en-erlang-ACH002706667-fr-pr>

Autres références :

- Transparents et documents fournis au cours
- Aide en ligne des logiciels utilisés.
- Sites d'information en ligne.