UCLouvain

linfo1104

2019

Paradigmes de programmation et concurrence

Au vu du contexte sanitaire lié à la propagation du coronavirus, les modalités d'organisation et d'évaluation des unités d'enseignement ont pu, dans différentes situations, être adaptées ; ces éventuelles nouvelles modalités ont été -ou seront- communiquées par les enseignant·es aux étudiant·es.

5 crédits	30.0 h + 30.0 h	Q2

Enseignants	Van Roy Peter ;
Langue d'enseignement	Français
Lieu du cours	Louvain-la-Neuve
Préalables	Ce cours suppose acquises des compétences de base en programmation visées par les cours LINFO1101 ou LEPL1401 et les notions sur l'algorithmique et les structures de données simples visées par le cours LEPL1402. *Le(s) prérequis de cette Unité d'enseignement (UE) sont précisés à la fin de cette fiche, en regard des programmes/formations qui proposent cette UE.*
Thèmes abordés	• Paradigmes de programmation (y compris fonctionnel, orienté objet, dataflow et multi-agent) • Sémantique formelle et techniques de raisonnement • Abstraction procédurale (programmation d'ordre supérieur) • Abstraction de données (y compris orienté objet et types de données abstraits) • Programmation concurrente (y compris état partagé, passage de messages et dataflow déterministe) • Programmation symbolique (y compris conception algorithmique)
Acquis d'apprentissage	1 Eu égard au référentiel AA du programme « Bachelier en Sciences de l'Ingénieur, orientation ingénieur civil », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants : • AA 1.1, 1.2 • AA 2.3, 2.4, 2.5, 2.6, 2.7 • AA 4.2, 4.3, 4.4 Eu égard au référentiel AA du programme « Bachelier en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants : • S1.I2, S1.I3, S1.I5 • S2.2, S2.3, S2.4 • S5.3, S5.4, S5.5 Les étudiants ayant suivi avec fruit ce cours seront capables de • spécifier les problèmes, les diviser dans leurs étapes de base, et concevoir des algorithmes et des abstractions pour les résoudre ; • choisir le bon paradigme de programmation et écrire un programme dans ce paradigme pour résoudre un problème ; • utiliser la sémantique formelle pour expliquer l'exactitude du programme ; • écrire de petits programmes concurrents dans le paradigme du dataflow déterministe. • penser à l'aide d'abstractions (raisonner correctement sur un système qui comprend plusieurs couches d'abstractions, et définir de nouvelles abstractions pour simplifier la résolution d'un problème) - - - - *La contribution de cette UE au développement et à la maîtrise des compétences et acquis du (des) programme(s) est accessible à la fin de cette fiche, dans la partie « Programmes/formations proposant cette unité d'enseignement (UE) ».*
Modes d'évaluation des acquis des étudiants	**En raison de la crise du COVID-19, les informations de cette rubrique sont particulièrement susceptibles d'être modifiées.** • Test dispensatoire 25% (vers la 7e semaine) • Projet 25% • Examen final (50%) (ou 75% si on refait la partie du test) Le projet est obligatoire et se fait pendant le quadrimestre. Il ne peut être fait qu'une fois et il compte pour toute l'année académique.

Méthodes d'enseignement	**En raison de la crise du COVID-19, les informations de cette rubrique sont particulièrement susceptibles d'être modifiées.** • Cours magistral chaque semaine • Séances de travaux pratiques en salle informatique chaque semaine, pour résoudre des problèmes simplifiés en utilisant les concepts vu au cours • Un grand projet de conception et d'implémentation pour appliquer ces concepts dans le cadre d'une application plus complexe.
Contenu	Le but de ce cours est d'élargir et approfondir les connaissances en programmation acquises dans les précédents cours. Ce cours abordera les sujets suivants: • Le cours donnera un cadre uniforme pour tous les concepts de programmation, organisés en paradigmes de programmation. • Le cours donnera une sémantique formelle et des techniques de raisonnement pour tous les paradigmes. • La programmation d'ordre supérieur sera utilisée comme principe organisateur pour construire des abstractions procédurales. • La programmation concurrente sera présentée avec ses trois formes principales, la concurrence à état partagé, la concurrence à passage de messages et la concurrence dataflow déterministe. • L'abstraction de données sera présentée dans sa forme générale et avec ses deux formes principales, la programmation orientée-objet et les types de données abstraits. • La programmation symbolique sera présentée avec la conception d'algorithmes symboliques. • Parmi les paradigmes de programmation abordés dans ce cours seront la programmation fonctionnelle, la programmation orientée objet, la programmation dataflow déterministe et la programmation multi-agent. Des exemples d'applications pratiques seront présentés pour tous les concepts et tous les paradigmes.
Ressources en ligne	LINFO1104 Moodle.
Bibliographie	Peter Van Roy et Seif Haridi, PROGRAMMATION: Concepts, techniques et modèles, Dunod, 2007 Peter Van Roy et Seif Haridi, Concepts, Techniques, and Models of Computer Programming, MIT Press, 2004
Autres infos	Prérequis: • Cours de programmation de seconde année Informatique 2 (LEPL1402) ou équivalent.
Faculté ou entité en charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)

Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Filière en Informatique	LINFO100P	5		
Bachelier en sciences informatiques	SINF1BA	5	LINFO1101	