

5.00 crédits

30.0 h + 30.0 h

Q2


**Cette unité d'enseignement n'est pas accessible aux étudiants d'échange !**

Enseignants	Van Roy Peter ;
Langue d'enseignement	Français
Lieu du cours	Charleroi
Préalables	<p>Ce cours suppose acquises des compétences de base en programmation visées par les cours LSINC1101/ LINFO1101 et les notions sur l'algorithmique et les structures de données simples visées par le cours LSINC1402/ LEPL1402.</p> <p><i>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.</i></p>
Thèmes abordés	<ul style="list-style-type: none"> <li>• Paradigmes de programmation (y compris fonctionnel, orienté objet, dataflow et multi-agent)</li> <li>• Sémantique formelle et techniques de raisonnement</li> <li>• Abstraction procédurale (programmation d'ordre supérieur)</li> <li>• Abstraction de données (y compris orienté objet et types de données abstraits)</li> <li>• Programmation concurrente (y compris état partagé, passage de messages et dataflow déterministe)</li> <li>• Programmation symbolique (y compris conception algorithmique)</li> </ul>
Acquis d'apprentissage	<p><b>A la fin de cette unité d'enseignement, l'étudiant est capable de :</b></p> <p>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 :</p> <ul style="list-style-type: none"> <li>• AA 1.1, 1.2</li> <li>• AA 2.3, 2.4, 2.5, 2.6, 2.7</li> <li>• AA 4.2, 4.3, 4.4</li> </ul> <p>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 :</p> <ul style="list-style-type: none"> <li>• S1.I2, S1.I3, S1.I5</li> <li>• S2.2, S2.3, S2.4</li> <li>• S5.3, S5.4, S5.5</li> </ul> <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> <li>• spécifier les problèmes, les diviser dans leurs étapes de base, et concevoir des algorithmes et des abstractions pour les résoudre ;</li> <li>• choisir le bon paradigme de programmation et écrire un programme dans ce paradigme pour résoudre un problème ;</li> <li>• utiliser la sémantique formelle pour expliquer l'exactitude du programme ;</li> <li>• écrire de petits programmes concurrents dans le paradigme du dataflow déterministe.</li> <li>• 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)</li> </ul>
Modes d'évaluation des acquis des étudiants	<ul style="list-style-type: none"> <li>• Test dispensatoire 25% (vers la 7e semaine)</li> <li>• Projet 25%</li> <li>• Examen final (50%) (ou 75% si on refait la partie du test)</li> </ul> <p>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.</p>
Méthodes d'enseignement	<ul style="list-style-type: none"> <li>• Cours magistral chaque semaine</li> <li>• Séances de travaux pratiques en salle informatique chaque semaine, pour résoudre des problèmes simplifiés en utilisant les concepts vu au cours</li> <li>• Un grand projet de conception et d'implémentation pour appliquer ces concepts dans le cadre d'une application plus complexe.</li> </ul>
Contenu	<p>Le but de ce cours est d'élargir et approfondir les connaissances en programmation acquises dans les précédents cours. Ce cours aborde les sujets suivants:</p>

	<ul style="list-style-type: none"><li>• Le cours donne un cadre uniforme pour tous les concepts de programmation, organisés en paradigmes de programmation.</li><li>• Le cours donne une sémantique formelle et des techniques de raisonnement pour tous les paradigmes présentés.</li><li>• Le cours donne une introduction au lambda calcul comme base de la programmation fonctionnelle et la programmation d'ordre supérieur.</li><li>• La programmation d'ordre supérieur est utilisée comme principe organisateur pour construire des abstractions procédurales.</li><li>• La programmation concurrente est présentée en deux formes, le dataflow déterministe et la concurrence à passage de messages.</li><li>• L'abstraction de données est présentée dans sa forme générale et avec ses deux formes principales, la programmation orientée-objet et les types de données abstraits.</li><li>• La programmation symbolique et la conception d'algorithmes sont utilisés partout dans le cours.</li><li>• Cinq paradigmes de programmation importants sont abordés dans ce cours: la programmation fonctionnelle, la programmation orientée objet, la programmation dataflow déterministe, les acteurs dataflow et la programmation avec objets actifs (multi-agent).</li></ul> <p>Des exemples d'applications pratiques sont présentés pour tous les concepts et tous les paradigmes.</p>
Faculté ou entité en charge:	SINC

<b>Programmes / formations proposant cette unité d'enseignement (UE)</b>				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Bachelier en sciences informatiques	SINC1BA	5	LSINC1101	