



5.00 crédits

30.0 h + 15.0 h

Q2

Enseignants	Pecheur Charles ;
Langue d'enseignement	Anglais > Facilités pour suivre le cours en français
Lieu du cours	Louvain-la-Neuve
Thèmes abordés	<ul style="list-style-type: none"> <li>• concepts de qualité du logiciel</li> <li>• spécification et exactitude d'un programme</li> <li>• test de logiciels</li> <li>• vérification et validation de logiciels</li> <li>• révision et audit de logiciels</li> <li>• métriques et mesures de logiciels</li> <li>• fiabilité de logiciels</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 « Master ingénieur civil en informatique », 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>• INFO2.3-5</li> <li>• INFO6.3</li> </ul> <p>Eu égard au référentiel AA du programme « Master [120] 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>• SIN1.M3</li> <li>• SIN2.3-5</li> <li>• SIN6.3</li> </ul> <p>Les étudiants ayant suivi ce cours avec fruit seront capables de :</p> <p>1</p> <ul style="list-style-type: none"> <li>• Définir la qualité des logiciels et décrire le rôle des activités d'assurance de la qualité dans le processus de logiciel.</li> <li>• Décrivez comment un contrat peut être utilisé pour spécifier le comportement d'un composant de programme.</li> <li>• Décrire et distinguer les différents types et niveaux de tests (unitaires, d'intégration, les systèmes, et l'acceptation).</li> <li>• Appliquer une variété de stratégies pour les tests et le débogage de programmes simples.</li> <li>• Décrivez comment les outils de vérification statique et dynamique disponibles peuvent être intégrés dans l'environnement de développement de logiciel.</li> <li>• Appliquer des techniques de spécification et d'analyse formelles à la conception de programmes avec une faible complexité.</li> <li>• Entreprendre une inspection d'un segment de code de taille moyenne.</li> <li>• Comparer les techniques simples de mesure de logiciels.</li> <li>• Décrire les approches pour l'estimation de fautes.</li> <li>• Expliquer les problèmes qui existent pour atteindre de niveaux élevés de fiabilité des logiciels.</li> </ul>
Modes d'évaluation des acquis des étudiants	Le cours inclut les devoirs, comptant pour 40% de la note, et un examen, comptant pour 60% de la note. Les devoirs ne peuvent pas être refaits pour la session de septembre; la note reste acquise en septembre.
Méthodes d'enseignement	<ul style="list-style-type: none"> <li>• Cours magistraux hebdomadaires</li> <li>• Séances d'exercices</li> <li>• Missions (par groupe de 2 étudiants)</li> </ul> <p>Les séances d'exercices et les missions sont synchronisées de manière à ce que les exercices préparent les étudiants à réaliser les missions.</p> <p>Selon les circonstances, tout ou partie des cours et des exercices pourraient être diffusés et enregistrés pour pouvoir être suivis à distance.</p>
Contenu	<ul style="list-style-type: none"> <li>• Principes de qualité logicielle</li> <li>• Modèles de logiciel</li> <li>• Test fonctionnel</li> </ul>

	<ul style="list-style-type: none"> <li>• Test structurel</li> <li>• Compléments de test</li> <li>• Exécution de tests</li> <li>• Analyse de programmes</li> <li>• Analyse à états finis</li> <li>• Fiabilité logicielle</li> <li>• Mesure de logiciel</li> </ul>
Ressources en ligne	<a href="https://moodleucl.uclouvain.be/course/view.php?id=10913">https://moodleucl.uclouvain.be/course/view.php?id=10913</a>
Bibliographie	<ul style="list-style-type: none"> <li>• Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement. Jeff Tian. 2005, Wiley-IEEE Computer Society Press.</li> <li>• M. Pezzè and Michal Young, Software Testing and Analysis: Process, Principles, and Techniques, Wiley, 2008.</li> <li>• J. Laski, W. Stanley. Software Verification and Analysis. Springer 2009.</li> <li>• N.E. Fenton and S.L. Pfleeger. Software Metrics: A Rigorous and Practical Approach. 2nd edition, Thomson Computer Press, 1996.</li> </ul>
Autres infos	<p>Préalables:</p> <ul style="list-style-type: none"> <li>• LEPL108, LINFO1104 : logique mathématique</li> <li>• LSINF1121 : maîtrise de la programmation orienté-objet, de l'algorithmique et des structures de données</li> <li>• LEPL1509 : avoir participé à un projet logiciel de taille réduite</li> </ul> <p>Avoir eu ou avoir simultanément une expérience avec le développement d'un systèmes informatique de taille moyenne à large</p>
Faculté ou entité en charge:	INFO

<b>Programmes / formations proposant cette unité d'enseignement (UE)</b>				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] : ingénieur civil biomédical	GBIO2M	5		
Master [120] : ingénieur civil en informatique	INFO2M	5		
Master [120] en sciences informatiques	SINF2M	5		