

5.00 crédits	30.0 h + 30.0 h	Q1
--------------	-----------------	----



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

Enseignants	Mens Kim ;Nijssen Siegfried ;
Langue d'enseignement	Français
Lieu du cours	Charleroi
Thèmes abordés	<ul style="list-style-type: none"> <li>• Introduction à la programmation ;</li> <li>• Langage de programmation Python ;</li> <li>• Analyse d'un problème informatique, conception, spécification et mise en oeuvre d'une solution ;</li> <li>• Structures de données linéaires ;</li> <li>• Concepts fondamentaux de la programmation orientée-objet.</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 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</li> <li>• S2.2, S2.4</li> </ul> <p>Les étudiants ayant suivi avec fruit ce cours seront capables de :</p> <ul style="list-style-type: none"> <li>• Appliquer les concepts et raisonnements informatiques à une problématique de complexité cadrée.</li> <li>• Décrire les outils, techniques, et calculs adéquats pour résoudre ce problématique disciplinaire cadrée.</li> <li>• Modéliser un problème et concevoir une ou plusieurs solutions techniques répondant au cahier des charges.</li> <li>• Implémenter et tester une solution sous la forme d'un prototype.</li> <li>• Collaborer en pairs ou en groupe et s'engager collectivement sur un plan de travail, un échéancier (et des rôles à tenir).</li> <li>• Communiquer sous forme graphique et schématique ; interpréter un schéma, présenter les résultats d'un travail, structurer des informations.</li> <li>• Lire, analyser et exploiter des documents techniques (normes, plans, cahier de charge, spécifications,...).</li> <li>• Rédiger des documents écrits de synthèse en tenant compte des exigences posées dans le cadre des missions (projets et problèmes).</li> <li>• Démontrer une bonne compréhension des concepts et de la méthodologie de la programmation, et de la programmation orientée-objet.</li> <li>• Utiliser à bon escient les éléments d'un langage de programmation tel que Python, y compris ces concepts orienté-objets.</li> </ul> <p>S1.I2 S2.2, S2.4</p>
Modes d'évaluation des acquis des étudiants	<p>Une mission de programmation est effectuée chaque semaine.</p> <p>Une interrogation intermédiaire a lieu en milieu de quadrimestre.</p> <p>L'examen en fin de quadrimestre a pour objectif de vérifier non seulement la connaissance de la matière, mais également la capacité à appliquer les connaissances acquises pour écrire de programmes Python simples, mais corrects. L'examen est fait sur ordinateur, sans accès à des ressources extérieures.</p> <p>La note finale du cours prend en compte l'interrogation intermédiaire et les missions durant le quadrimestre, en plus de la note de l'examen.</p> <p>Les missions et l'interrogation ne peuvent pas être refaits pour les sessions de juin ou septembre.</p> <p>Si la note de l'interrogation intermédiaire est supérieure à celle de l'examen, elle interviendra pour 1/3 et la note de l'examen pour 2/3.</p> <p>Si la note de l'interrogation intermédiaire est inférieure à celle de l'examen, seule la note de l'examen sera utilisée pour calculer la note finale.</p> <p>Un bonus de 1 point sera accordé aux étudiants ayant participé de manière autonome et soumis régulièrement les missions de programmation pendant le quadrimestre. L'utilisation d'aide extérieure ou d'outils d'IA générative dans les missions n'est pas souhaitée.</p> <p>En cas de détection confirmée de plagiat lors des examens, les titulaires du cours se réservent le droit de demander à l'étudiant de passer une interrogation orale.</p>

<p>Méthodes d'enseignement</p>	<p>Les méthodes d'enseignement utilisées encouragent l'apprentissage actif des l'étudiants, par :</p> <ul style="list-style-type: none"> <li>• des cours magistraux,</li> <li>• des séances d'exercices encadrées par un tuteur,</li> <li>• de la programmation sur la plateforme INGINIOUS.</li> </ul> <p>Même si une préférence est donné au séances en présentiel, en fonction de la situation sanitaire et du nombre d'étudiants inscrits une autre forme d'enseignement et de l'évaluation (distanciel, co-modal ou hybride) peut être envisagée.</p>
<p>Contenu</p>	<ul style="list-style-type: none"> <li>• Programmes, code source, exécution de programmes ;</li> <li>• Identifiants, variables, valeurs, types, affectation ;</li> <li>• Expressions, instructions ;</li> <li>• Structures conditionnelles et boucles ;</li> <li>• Fonctions, paramètres, appels, exécution, résultats, portée de variables ;</li> <li>• Spécifications, tests ;</li> <li>• Modules ;</li> <li>• Structures de données, listes, chaînes de caractères et leurs opérations ;</li> <li>• Références et structures de données imbriquées ;</li> <li>• Listes imbriquées, tuples, matrices, et dictionnaires ;</li> <li>• Algorithmes de recherche dichotomiques ;</li> <li>• Manipulation de fichiers, entrées-sorties ;</li> <li>• Mécanismes d'exception ;</li> <li>• La programmation orientée objets, ramasse-miettes ;</li> <li>• Classes, objets, constructeurs, méthodes ;</li> <li>• Références vers un objet, références à et appeler soi-même ;</li> <li>• Variables de classe, d'instance, locales, leur portée et visibilité ;</li> <li>• Composition d'une classe, héritage et encapsulation ;</li> <li>• Polymorphisme, appels à la classe mère (super) et la liaison dynamique des méthodes ;</li> <li>• Egalité d'objets ;</li> <li>• Structure de données chaînées.</li> </ul>
<p>Ressources en ligne</p>	<p>Tout le matériel du cours sera disponible en ligne : transparents, syllabus, exercices, ...</p>
<p>Faculté ou entité en charge:</p>	<p>SINC</p>

<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		