

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

Enseignants	Mens Kim ;Nijssen Siegfried ;
Langue d'enseignement	Français
Lieu du cours	Charleroi
Préalables	Néant
Thèmes abordés	<ul style="list-style-type: none"> • Introduction à la programmation ; • Langage de programmation Python ; • Analyse d'un problème informatique, conception, spécification et mise en oeuvre d'une solution ; • Structures de données linéaires ; • Concepts fondamentaux de la programmation orientée-objet.
Acquis d'apprentissage	<p>A la fin de cette unité d'enseignement, l'étudiant est capable de :</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"> • S1.I2 • S2.2, S2.4 <p>Les étudiants ayant suivi avec fruit ce cours seront capables de :</p> <ul style="list-style-type: none"> • Appliquer les concepts et raisonnements informatiques à une problématique de complexité cadrée. • Décrire les outils, techniques, et calculs adéquats pour résoudre ce problématique disciplinaire cadrée. • Modéliser un problème et concevoir une ou plusieurs solutions techniques répondant au cahier des charges. • Implémenter et tester une solution sous la forme d'un prototype. • Collaborer en pairs ou en groupe et s'engager collectivement sur un plan de travail, un échéancier (et des rôles à tenir). • Communiquer sous forme graphique et schématique ; interpréter un schéma, présenter les résultats d'un travail, structurer des informations. • Lire, analyser et exploiter des documents techniques (normes, plans, cahier de charge, spécifications,...). • Rédiger des documents écrits de synthèse en tenant compte des exigences posées dans le cadre des missions (projets et problèmes). • Démontrer une bonne compréhension des concepts et de la méthodologie de la programmation, et de la programmation orientée-objet. • Utiliser à bon escient les éléments d'un langage de programmation tel que Python, y compris ces concepts orienté-objets. <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.</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 et soumis régulièrement les missions de programmation pendant le quadrimestre.</p> <p>En cas de détection de plagiat confirmé par un outil de plagiat 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"> • des cours magistraux, • des séances d'exercices encadrées par un tuteur, • de la programmation sur la plateforme INGINIOUS. <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"> • Programmes, code source, exécution de programmes ; • Identifiants, variables, valeurs, types, affectation ; • Expressions, instructions ; • Structures conditionnelles et boucles ; • Fonctions, paramètres, appels, exécution, résultats, portée de variables ; • Spécifications, tests ; • Modules ; • Structures de données, listes, chaînes de caractères et leurs opérations ; • Références et structures de données imbriquées ; • Listes imbriquées, tuples, matrices, et dictionnaires ; • Algorithmes de recherche dichotomiques ; • Manipulation de fichiers, entrées-sorties ; • Mécanismes d'exception ; • La programmation orientée objets, ramasse-miettes ; • Classes, objets, constructeurs, méthodes ; • Références vers un objet, références à et appeler soi-même ; • Variables de classe, d'instance, locales, leur portée et visibilité ; • Composition d'une classe, héritage et encapsulation ; • Polymorphisme, appels à la classe mère (super) et la liaison dynamique des méthodes ; • Egalité d'objets ; • Structure de données chaînées.
<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>EPL</p>

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