

**Haute École Libre de Bruxelles – Ilya Prigogine**  
**1 DESCRIPTION DES UNITÉS D'ENSEIGNEMENT**

**PROGRAMMATION I**

**CODE : UE1-2**

<b>Catégorie : ÉCONOMIQUE</b>	
Section : <b>INFORMATIQUE DE GESTION</b>	Sous-section / Finalité / Option : <b>Sans objet</b>
Implantation : <b>Campus Jupiter</b> , Avenue Jupiter, 201 – 1190 Bruxelles Téléphone secrétariat : 02 / 340 16 70	
Cycle :  Bloc d'études : <b>1</b> Situation dans la formation : <input checked="" type="checkbox"/> 1 <sup>er</sup> quadrimestre <input type="checkbox"/> 2 <sup>ème</sup> quadrimestre Niveau du cadre européen des certifications :	Unité d'enseignement pré-requise : <b>Néant</b> Unité d'enseignement co-requise : <b>Néant</b> Volume horaire : <b>162h</b> Nombre de crédits ECTS : <b>14</b> <b>Obligatoire / Optionnelle</b> Langue d'enseignement : <b>français</b> Langue d'évaluation : <b>français</b>
<u>Responsable de l'UE</u> : <b>Monsieur Etienne PUGNAGHI</b>	<u>Titulaires des activités d'apprentissage</u> : <b>Madame Imen BEN HNIA</b> ( <i>Java</i> ) <b>Monsieur Etienne PUGNAGHI</b> ( <i>Java</i> ) <b>Monsieur Olivier PONCHAUT</b> ( <i>Java</i> ) <b>Monsieur Jacques MYTER</b> ( <i>Méthodologie de la programmation</i> )
<b>CONTRIBUTION AU PROFIL D'ENSEIGNEMENT :</b>	
En regard de l'ensemble du programme de formation, l'UE contribue au développement des compétences et capacités suivantes :	
<b><u>Compétences</u></b> :	
C3 : Mobiliser les savoirs et savoir-faire propres à l'informatique de gestion	
<b><u>Capacités</u></b> :	
C3 <ul style="list-style-type: none"> <li>• Concevoir, implémenter et maintenir des algorithmes répondant aux spécifications et fonctionnalités fournies</li> <li>• Concevoir, implémenter, administrer et utiliser avec maîtrise un ensemble structuré de données</li> </ul>	

## ACQUIS D'APPRENTISSAGE SPÉCIFIQUES

### **De manière générale pour l'unité d'enseignement :**

Au terme de cette unité d'enseignement, l'apprenant sera capable de déterminer un choix judicieux du type de variables à utiliser, de faire apparaître la structure d'un algorithme, de le rédiger et le tester en Java.

### **Par activité d'apprentissage :**

#### **En Méthodologie de la programmation:**

Au terme de l'activité d'apprentissage, l'étudiant sera capable de faire apparaître la structure algorithmique de tout problème dont il connaît la méthode de résolution et de la rédiger dans un langage structuré (LDA), proche des langages de programmation du type Java, C#, ... Les « objets » que l'étudiant sera capable de manipuler sont : les nombres, les chaînes de caractères, les tableaux, les fichiers séquentiels et les pointeurs.

#### **En Java:**

Au terme de l'activité d'apprentissage, l'étudiant sera capable de mobiliser les compétences acquises en Méthodologie de la programmation pour la rédaction d'un programme dans un langage de haut niveau (Java).

## CONTENU SYNTHÉTIQUE

### **En Méthodologie de la programmation:**

- Variables numériques, booléennes, chaînes de caractères, pointeurs
- Structures de base : séquence, alternative, répétitive
- Procédures et fonctions (passage par valeur et adresse)
- Tableaux et fichiers séquentiels
- Pointeurs

### **En Java:**

- Structure de base d'un programme
- Déclaration de variables et types de base
- Structures de contrôle et boucles
- Tableaux
- Méthodes
- Fichiers séquentiels
- Création et utilisation d'objets

## MÉTHODES D'APPRENTISSAGE

### **En Méthodologie de la programmation:**

Présentation par l'enseignant du formalisme utilisé pour décrire les différents types de structures et de données. Ensuite, exercices dirigés en classe et exercices préparatoires à faire à domicile.

### **En Java:**

Présentation de la syntaxe Java par l'enseignant suivie de la réalisation d'exercices pratiques à partir d'exercices en LDA (=Langage de Description d'Algorithme) rédigés en méthodologie ou/et d'énoncés textuels. L'implémentation en Java de la solution sera obligatoirement précédée de l'écriture de cet algorithme en LDA.

SUPPORTS DE COURS		
	Obligatoire	En ligne
Méthodologie de la programmation	Oui	Oui
Java	Oui	Oui
MODALITÉS D'ÉVALUATION		
<p><b>Type d'évaluation :</b> Epreuve intégrée de type examen écrit (100%) remédiable (à 100%) à chaque session.</p> <p><b>Calcul de la note de l'UE :</b> La note de l'unité d'enseignement vaut la note de l'épreuve intégrée.</p>		
SOURCES DOCUMENTAIRES		
<p><u>Utilisées par l'enseignant :</u></p> <p>DARWIN (Ian F.), <i>Java en action</i>, O'REILLY, 2002  DELANNOY (C.), <i>Programmer en Java</i>, Eyrolles, 2014  FORISEK (M.) &amp; STEINOVA (M.), <i>Explaining Algorithms Using Metaphors</i>, Springer, 2013  <b>GROUSSARD (T.)</b>, <i>JAVA 7 : Les fondamentaux du langage Java</i>, Eni, 2011  HROMKOVIC (J.), <i>Algorithmic Adventures, From Knowledge to Magic</i>, Springer, 2009  LIANG PEARSON (Y.), <i>Introduction to Java Programming</i>, Prentice Hall, 2013  <b>ROHAUT (S.)</b>, <i>Algorithmique : Techniques fondamentales de programmation (Exemples en Java)</i>, Eni, 2013  SHEN (A.), <i>Algorithms and Programming (Problems and Solutions)</i>, Springer, 2010</p> <p><u>Proposées à l'appui du travail personnel de l'étudiant :</u></p> <p><b>GROUSSARD (T.)</b>, <i>JAVA 7 : Les fondamentaux du langage Java</i>, Eni, 2011  KNUTH (D. E.), <i>The art of computer programming : Fundamental algorithms</i>, Addison-Wesley Professional, 1997  LIANG PEARSON (Y.), <i>Introduction to Java Programming</i>, Prentice Hall, 2013  SEGEWICK (R.), WAYNE (Kevin), <i>Algorithms</i>, Pearson, 2011  SPRANKLE (M.), HUBBARD (J.), <i>Problem solving and programming concepts</i>, Pearson, 2012  VOCKING (B.), ALT (H.) Alt, DIETZFELBINGER (M.), REISCHUK (R.), SCHEILDELER (C.), VOLLMER (H.), WAGNER (D.), <i>Algorithms Unplugged</i>, Springer, 2011</p> <p>Supports de cours et autres :  <a href="https://ecampus.helb-prigogine.be">https://ecampus.helb-prigogine.be</a></p>		