

Bachelier en informatique orientation technologie de l'informatique

HELHa Tournai - Frinoise Rue Frinoise 12 7500 TOURNAI

Tél : +32 (0) 69 89 05 60

Fax : +32 (0) 69 89 05 65

Mail : tech.tournai@helha.be

1. Identification de l'Unité d'Enseignement

UE2202 Programmation			
Ancien Code	TEIC2B11	Caractère	Obligatoire
Nouveau Code	XIIT2110		
Bloc	2B	Quadrimestre(s)	Q2
Crédits ECTS	6 C	Volume horaire	77 h
Coordonnées des responsables et des intervenants dans l'UE	Frédéric PLUQUET (frederic.pluquet@helha.be)		
Coefficient de pondération	60		
Cycle et niveau du Cadre Francophone de Certification	bachelier / niveau 6 du CFC		
Langue d'enseignement et d'évaluation	Français		

2. Présentation

Introduction

Cette unité d'enseignement vise à rendre le bachelier en Informatique capable de gérer des activités ou des projets techniques ou professionnels complexes, en faisant preuve de responsabilité dans la prise de décisions dans des contextes professionnels ou d'études imprévisibles. Ils seront aussi amenés à prendre des responsabilités en matière de développement professionnel individuel et collectif. Conception, réalisation, réglage, programmation.

Contribution au profil d'enseignement (cf. référentiel de compétences)

Cette Unité d'Enseignement contribue au développement des compétences et capacités suivantes :

Compétence 1 **Communiquer et informer**

- 1.1 Choisir et utiliser les moyens d'information et de communication adaptés

Compétence 2 **Collaborer à la conception, à l'amélioration et au développement de projets**

- 2.1 Elaborer une méthodologie de travail
 2.2 Planifier des activités et évaluer la charge et la durée de travail liées à une tâche
 2.3 Analyser une situation donnée sous ses aspects techniques et scientifiques

Compétence 4 **S'inscrire dans une démarche de respect des réglementations**

- 4.3 Respecter les prescrits légaux relatifs au contexte dans lequel s'exerce l'activité (exemple code du bien-être au travail, RGPD, le droit à l'image, licences logiciels ...)

Acquis d'apprentissage visés

Programmation :

A la fin de cette activité d'apprentissage, l'étudiant doit être capable de :

- Utiliser les concepts de développement d'applications dans un environnement graphique multi-plateforme en utilisant le langage Java
- Réaliser de la programmation événementielle en utilisant le langage Java
- Savoir gérer une architecture de code (MVC, ...) en y intégrant des design patterns
- Savoir gérer une base de données au travers d'un programme Java
- Editer et lire des fichiers textes et binaires en utilisant le langage Java

Laboratoire de programmation :

- Lors des séances de laboratoire de programmation et la fin de cette partie, l'étudiant devra être capable, à partir d'un énoncé donné, de créer des programmes avec interfaces graphiques mettant en œuvre les différentes notions théoriques vues dans la partie programmation sur machine en utilisant ses notes.

- Lors des séances de laboratoire de programmation et pour la fin de cette partie, l'étudiant devra être capable d'analyser, de créer un projet en rapport avec les différentes notions vues sur un sujet fixé et en travaillant en groupe.

Liens avec d'autres UE

Prérequis pour cette UE : aucun

Corequis pour cette UE : aucun

3. Description des activités d'apprentissage

Cette unité d'enseignement comprend l(es) activité(s) d'apprentissage suivante(s) :

TEIC2B11A	Programmation	36 h / 3 C
-----------	---------------	------------

TEIC2B11B	Laboratoire de programmation (projet seul 3h)	41 h / 3 C
-----------	---	------------

Les descriptions détaillées des différentes activités d'apprentissage sont reprises dans les fiches descriptives jointes.

4. Modalités d'évaluation

Les 60 points attribués dans cette UE sont répartis entre les différentes activités de la manière suivante :

TEIC2B11A	Programmation	30
-----------	---------------	----

TEIC2B11B	Laboratoire de programmation (projet seul 3h)	30
-----------	---	----

Les formes d'évaluation et les dispositions complémentaires particulières des différentes activités d'apprentissage sont reprises dans les fiches descriptives jointes.

Dispositions complémentaires relatives à l'UE

La note de cette unité d'enseignement est obtenue en effectuant une moyenne géométrique pondérée des notes finales obtenues lors des évaluations des différentes activités d'apprentissage qui la composent.

Si une des cotes de AA est inférieure à 8/20, elle devient la cote de l'UE.

Si une AA est $<$ à 8/20, la moyenne géométrique pondérée ne se fait pas et cette note en échec devient la note de l'UE.

Si plusieurs AA sont $<$ à 8/20, la moyenne géométrique pondérée ne sera pas appliquée, la note de l'UE sera la note de l'AA la plus basse.

D'autres modalités d'évaluation peuvent être prévues en fonction du parcours académique de l'étudiant. Celles-ci seront alors consignées dans un contrat didactique spécifique proposé par le responsable de l'UE, validé par la direction ou son délégué et signé par l'étudiant pour accord.

5. Cohérence pédagogique

Néant

Référence au RGE

En cas de force majeure, une modification éventuelle en cours d'année peut être faite en accord avec le Directeur de département, et notifiée par écrit aux étudiants. (article 66 du règlement général des études 2024-2025).

Bachelier en informatique orientation technologie de l'informatique

HELHa Tournai - Frinoise Rue Frinoise 12 7500 TOURNAI
 Tél : +32 (0) 69 89 05 60 Fax : +32 (0) 69 89 05 65 Mail : tech.tournai@helha.be

1. Identification de l'activité d'apprentissage

Programmation			
Ancien Code	24_TEIC2B11A	Caractère	Obligatoire
Nouveau Code	TIIT2111		
Bloc	2B	Quadrimestre(s)	Q2
Crédits ECTS	3 C	Volume horaire	36 h
Coordonnées du Titulaire de l'activité et des intervenants	Frédéric PLUQUET (frederic.pluquet@helha.be)		
Coefficient de pondération	30		
Langue d'enseignement et d'évaluation	Français		

2. Présentation

Introduction

Cette activité d'apprentissage vise à rendre le bachelier en Informatique capable de gérer des activités techniques ou professionnelles complexes, en faisant preuve de responsabilité dans la prise de décisions dans des contextes professionnels ou d'études imprévisibles. Ils seront aussi amenés à prendre des responsabilités en matière de développement professionnel individuel. Conception, réalisation, réglage, programmation.

Objectifs / Acquis d'apprentissage

A la fin de cette activité d'apprentissage, l'étudiant doit être capable de :

- Utiliser les concepts de développement d'applications dans un environnement graphique multi-plateforme en utilisant le langage Java
- Réaliser de la programmation événementielle en utilisant le langage Java
- Savoir faire communiquer plusieurs programmes Java en réseau
- Editer et lire des fichiers textes et binaires en utilisant le langage Java.

3. Description des activités d'apprentissage

Contenu

1. IDE IntelliJ
2. Interpréteur et machine virtuelle
3. Différences avec le C++
4. Héritage simple, interfaces et polymorphisme
5. Method binding : overloading, overriding, this et super
6. Classes anonymes, internes, interface fonctionnelle, lambdas, annotations
7. Fichiers et Streams
8. MVC simple
9. JavaFX
 1. présentation générale
 2. l'éditeur graphique (SceneBuilder)
 3. la scène
 4. les composants de base
 5. programmation événementielle
 6. les layouts
 7. MVC, le data binding
 8. déploiement
10. Lien avec une base de données

Démarches d'apprentissage

Cours théorique magistral comprenant de nombreux exemples commentés ainsi que des exercices à réaliser en classe. La résolution des exercices s'effectue de manière interactive avec les étudiants. L'expertise enseignée est reliée à un contexte d'utilisation (méthode cognitive).

Dispositifs d'aide à la réussite

Réalisation seul d'exercices en classe sur la matière vue et correction collective de ces exercices. Au travers de l'auto-constructivisme, les étudiants sont amenés à prendre du recul sur une solution donnée et d'en analyser les points forts, les points faibles et les améliorations possibles à apporter.

Sources et références

Object-oriented software engineering practical software development using UML and Java, <https://dl.acm.org/citation.cfm?id=1594049>

Supports en ligne

Les supports en ligne et indispensables pour acquérir les compétences requises sont :

Exemples et exercices disponibles sur le Moodle de l'AA.

4. Modalités d'évaluation

Principe

Q2:

- Interrogation sur papier sans notes et/ou sur machine. (20%)
- Examen écrit sans notes. (80%)

Q3 :

- Examen écrit sans les notes. (100%)

Pondérations

	Q1		Q2		Q3	
	Modalités	%	Modalités	%	Modalités	%
production journalière			Int	20		
Période d'évaluation			Exe	80	Exe	100

Int = Interrogation(s), Exe = Examen écrit

La pondération de cette activité d'apprentissage au sein de l'UE dont elle fait partie vaut 30

Dispositions complémentaires

En cas d'échec à l'examen de Q2, l'examen pourra être représenté en Q3, et l'interrogation ne sera pas conservée. Un certificat médical entraîne, au cours de la même session, la représentation d'une épreuve similaire (dans la mesure des possibilités d'organisation).

Référence au RGE

En cas de force majeure, une modification éventuelle en cours d'année peut être faite en accord avec le Directeur de département, et notifiée par écrit aux étudiants. (article 66 du règlement général des études 2024-2025).

Bachelier en informatique orientation technologie de l'informatique

HELHa Tournai - Frinoise Rue Frinoise 12 7500 TOURNAI
 Tél : +32 (0) 69 89 05 60 Fax : +32 (0) 69 89 05 65 Mail : tech.tournai@helha.be

1. Identification de l'activité d'apprentissage

Laboratoire de programmation (projet seul 3h)			
Ancien Code	24_TEIC2B11B	Caractère	Obligatoire
Nouveau Code	TIIT2112		
Bloc	2B	Quadrimestre(s)	Q2
Crédits ECTS	3 C	Volume horaire	41 h
Coordonnées du Titulaire de l'activité et des intervenants	Frédéric PLUQUET (frederic.pluquet@helha.be)		
Coefficient de pondération	30		
Langue d'enseignement et d'évaluation	Français		

2. Présentation

Introduction

Cette activité d'apprentissage vise à rendre le bachelier en Informatique capable de gérer des activités techniques ou professionnelles complexes, en faisant preuve de responsabilité dans la prise de décisions dans des contextes professionnels ou d'études imprévisibles. Ils seront aussi amenés à prendre des responsabilités en matière de développement professionnel individuel. Conception, réalisation, réglage, programmation.

Objectifs / Acquis d'apprentissage

- Lors des séances de laboratoire de programmation et la fin de cette partie, l'étudiant devra être capable, à partir d'un énoncé donné, de créer des programmes avec interfaces graphiques mettant en œuvre les différentes notions théoriques vues dans l'activité d'apprentissage programmation sur machine en utilisant ses notes.
- Lors des séances de laboratoire de programmation et pour la fin de cette partie, l'étudiant devra être capable d'analyser, de créer un projet en rapport avec les différentes notions vues sur un sujet fixé et en travaillant en groupe

3. Description des activités d'apprentissage

Contenu

Le laboratoire consistera en des exercices à résoudre en classe, en rapport avec la matière vue dans l'ActAp TEIT2B11A.

Ce seront donc des applications écrites en Java.

De plus certaines pratiques de bonne qualité de code seront introduites comme:

les tests unitaires;

la documentation javadoc ...

Les labos sont divisés en 2 parties : une partie d'exercices et une partie projet durant laquelle les étudiants mettent en œuvre leur connaissance sur un projet conséquent.

Démarches d'apprentissage

Les étudiants auront à développer leurs exercices en groupes, et de manière la plus autonome possible. Pour cela, le dialogue est ouvert entre les enseignants et les étudiants;

l'accès à Internet est donné en permanence aux étudiants.

Durant tout le laboratoire, des explications sont données de manière orale, en fonction des besoins et difficultés rencontrées par les étudiants.

Réalisation d'un projet final

Afin de concrétiser et de faire une synthèse de l'ensemble des notions abordées dans l'UE, l'étudiant devra développer un projet final répondant à un cahier des charges précis.

Dispositifs d'aide à la réussite

Les différents énoncés des laboratoires seront:

- soit corrigés en classe, par la présentation d'une solution possible;
- soit de la remise aux étudiants d'une série de fichiers informatiques illustrant une solution possible.

Sources et références

- Object-oriented software engineering practical software development using UML and Java, <https://dl.acm.org/citation.cfm?id=1594049>

Supports en ligne

Les supports en ligne et indispensables pour acquérir les compétences requises sont :

L'ensemble des énoncés, références, notes... seront disponibles sur le site Connected de l'AA.

4. Modalités d'évaluation

Principe

L'évaluation se fera de la manière suivante:

- un projet final reprenant l'ensemble des notions abordées (60%);
- un examen final (40%)

L'examen final consistera à refaire et/ou à expliquer un partie du projet final et/ou des laboratoires.

La note finale sera calculée de la manière suivante:

- en cas de note à l'examen final strictement inférieure à 8, cette note sera la note finale (note absorbante);
- sinon, la note finale sera calculée selon la pondération donnée ci-dessus.

Pondérations

	Q1		Q2		Q3	
	Modalités	%	Modalités	%	Modalités	%
production journalière			Prj	60	Prj	60
Période d'évaluation			Exp	40	Exp	40

Prj = Projet(s), Exp = Examen pratique

La pondération de cette activité d'apprentissage au sein de l'UE dont elle fait partie vaut 30

Dispositions complémentaires

Pour le Q3, l'étudiant devra remettre un projet final. La pondération sera :

- 60% pour le projet final
- 40% pour l'examen final.

La note finale sera calculée de la manière suivante:

- en cas de note à l'examen final strictement inférieure à 8, cette note sera la note finale (note absorbante);
- sinon, la note finale sera calculée selon la pondération donnée ci-dessus.

Référence au RGE

En cas de force majeure, une modification éventuelle en cours d'année peut être faite en accord avec le Directeur de département, et notifiée par écrit aux étudiants. (article 66 du règlement général des études 2024-2025).