Programmation d'applications distribuées et en réseau

TEM1005-NAIN_

Domaine : **Sciences** Grade académique : **Master en architecture des systèmes informatiques - Namur**
Niveau de certification : **7** Cycle : **Deuxième cycle** Année académique : **2026-2027** (Q1)

CRÉDITS ECTS	VOLUME HORAIRE
4	**50 h**
PRÉREQUIS	**CO-REQUIS**
Aucun	*Aucun*

IMPLANTATION	DIRECTEUR DE DÉPARTEMENT DU DOMAINE	RESPONSABLE DE L'UE	ENSEIGNANTS INTERVENANTS
Département technique de Namur	Willem Kuypers	Laurent PHILIPPE	Adrien HUYGENS Laurent PHILIPPE

PROGRAMME D'ÉTUDES

Programme	Bloc	Caractère
IE-Master en architecture des systèmes informatiques	1	Obligatoire

Partie 1 Contribution de l'UE au profil d'enseignement du programme et acquis d'apprentissage spécifiques (AAS) sanctionnés par l'évaluation

CONTEXTUALISATION

UE prérequise : aucune

Cette UE contribue à l'atteinte des acquis d'apprentissage terminaux suivants :

• Résoudre selon une approche analytique et systémique des problèmes liés à des situations nouvelles ou présentant un certain degré d'incertitude

– Identifier, traiter et synthétiser les données pertinentes

– Rechercher les ressources nécessaires

– Transposer les résultats des concepts à la problématique ou situation traitée

– Argumenter les choix proposés

• Contribuer à des travaux de recherche et de développement

– Analyser une problématique de manière critique

– Identifier et proposer des solutions innovantes et/ou améliorations

• Communiquer dans un cadre professionnel avec les collaborateurs, les clients

– Rédiger des rapports, des cahiers des charges, fiches techniques ou manuels

– S'exprimer de manière adaptée en fonction du public

– Utiliser efficacement les outils de communication adéquats

• Agir de façon réflexive et autonome, en équipe, en partenariat

– Organiser son temps, respecter les délais

– Acquérir une capacité d'autocritique

– Actualiser ses connaissances et compétences

– Collaborer activement avec d'autres

– Animer et accompagner une équipe

– Assumer les responsabilités associées aux actes posés

• Innover, concevoir ou optimiser un système

– Intégrer l'ensemble des composantes d'un système à partir des résultats d'analyse

– Concevoir des procédures et des dispositifs

– Modéliser, dimensionner et optimiser des systèmes

• Gérer les ressources techniques et financières de systèmes

– Assurer un suivi

– Evaluer les processus et les résultats

– Proposer et appliquer les actions correctives

• Utiliser des procédures et des outils

– Exploiter les outils appropriés afin de résoudre une tâche spécifique

– Effectuer des tests, des contrôles et des mesures

– Effectuer des paramétrages, des réglages

– Respecter la déontologie et les règlementations en vigueur dans la profession

ACQUIS D'APPRENTISSAGE SPÉCIFIQUES (AAS)

À la fin de cette UE, l'étudiant sera capable :

• d'identifier les différents types d'architecture distribuée et de sélectionner celle qui correspond au mieux à un besoin,

• de suivre une approche SCRUM adaptée au développement :

– Jeu du planning,

– Organisation et planification de sprints,

– Gestion des interactions avec le client et avec les collaborateurs ;

• de concevoir une application :

– Conception et rédaction des spécifications sous forme de User Stories,

– Conception, analyse et évaluation de performances d'algorithmes, et plus précisément d'algorithmes distribués,

– Réfléchir à la manière optimale d'ajouter une fonctionnalité (structure de données, algorithme, protocole, accès concurrents, ...) ;

• de gérer un projet de manière efficace et pérenne :

– Test driven development,

– Intégration continue,

– Utilisation d'un CVS (git) pour la gestion du projet ;

• de développer en Python une application :

– multithread,

– utilisant des sockets,

– gérant de manière efficace les erreurs d'exécution,

– communicant de manière sécurisée.

OBJECTIFS DE DÉVELOPPEMENT DURABLE

Document annexe à la fiche UE accessible via un lien explicitant en quoi le programme d'études s'inscrit dans une politique de développement durable. Une ou plusieurs UE spécifique(s) pourraient être ciblée(s) dans cette annexe.

Partie 2 Description de l'unité d'enseignement

OBJECTIFS

L'objectif du cours est de présenter certains principes et méthodes de base de génie logiciel (méthodes agiles) et de les appliquer dans le cadre spécifique du développement d'applications distribuées et en réseau. L'UE s'articule autour d'un projet visant à la livraison d'une application écrite en Python et dont le développement devra suivre scrupuleusement les techniques d'organisation et de gestion de projets vues en cours (SCRUM).

CONTENU

Théorie

Les séances de cours théorique aborderont les points suivants :

• Introduction Agile (principes et fonctionnement)

• Analyse dépôt git (implémentation agile)

• Gestion des plannings + introduction aux tests unitaires

• Écriture de tests unitaires sur code existant

• Test Driven Development + exercices

• Threading et socket

• TDD avec utilisation de Mock

• Algorithmique distribuée

• TDD sur algorithmes distribués

Un accent particulier sera apporté aux bonnes pratiques de développement en Python (gestion des exceptions et des cas problématiques notamment) tout au long des séances de théorie, sans que ces dernières ne fassent l'objet d'une présentation à part entière, l'utilisation de python étant considérée comme un prérequis.

SOURCES ET RÉFÉRENCES

À définir

SUPPORTS DE COURS

Activité d'apprentissage	Support
AAIN1005-A — Théorie	Sur la plateforme d'e-learning : - Slides des cours théoriques - Énoncé des exercices et solutions
AAIN1005-B — Laboratoires	*À définir*

Partie 3 Méthodologie mise en œuvre et éventuel découpage de l'UE

DESCRIPTION DES ACTIVITÉS D'APPRENTISSAGE (AA) AU REGARD DE LEUR STATUT

Activité d'apprentissage	Heures	Langue d'enseignement	Langue d'évaluation
AAIN1005-A — Théorie	30 h	français mais une compréhension à la lecture de l'anglais peut être nécessaire (exemples et références)	français / anglais (commentaires personnels dans le code)
AAIN1005-B — Laboratoires	20 h	Français	Français

MÉTHODES D'ENSEIGNEMENT ET D'APPRENTISSAGE

Activité d'apprentissage	Méthodes d'enseignement et d'apprentissage
AAIN1005-A — Théorie	Le cours théorique repose sur une présentation ex cathedra des principes et outils, suivie par la mise en pratique au travers d L'étudiant est tenu de maîtriser le contenu des présentations ex cathedra lors des séances d'exercices pour profiter pleinement La syntaxe de Python ainsi que les bonnes pratiques de base (gestion des erreurs, des exceptions, des cas particuliers, ...) so Les séances de cours théorique aborderont les points suivants : - Introduction Agile (principes et fonctionnement) - Analyse dépôt git (implémentation agile) - Gestion des plannings + introduction aux tests unitaires - Écriture de tests unitaires sur code existant - Test Driven Development (TDD) + exercices - Threading et socket - TDD avec utilisation de Mock - Algorithmique distribuée - TDD sur algorithmes distribués Un accent particulier sera apporté aux bonnes pratiques de développement en Python (gestion des exceptions et des cas problémat
AAIN1005-B — Laboratoires	*À définir*

DÉTERMINATION DE LA COHÉRENCE PÉDAGOGIQUE EN CAS DE REGROUPEMENT D'AA FAISANT L'OBJET D'UNE ÉVALUATION DISTINCTE

Sans objet

Partie 4 Évaluation de l'unité d'enseignement

STRUCTURE DU DISPOSITIF GÉNÉRAL D'ÉVALUATION (EN LIEN AVEC LE PROGRAMME D'ÉTUDES)

Évaluation intégrée : oui

UE remédiable.

Évaluation : session de janvier (Q1) et seconde session (Q3).

Le travail se faisant en groupe, il sera apporté une attention particulière à la bonne compréhension du code fourni : l'incapacité à expliquer tout ou partie de ce dernier entraînera un échec.

L'évaluation est une évaluation par acquis d'apprentissage, la non validation d'un de ces acquis peut entraîner un échec.

MÉTHODOLOGIE DE CONSTRUCTION DE LA NOTE FINALE DE L'UE

Septembre - Janvier et deuxième session

• Travaux : Projet : qualité du code fourni - pondération : 50 %

• Supervision : Projet : application de la méthodologie SCRUM - pondération : 50 %

Un projet sera réalisé en suivant la méthodologie SCRUM adaptée au développement d'applications. Le bon suivi de la méthodologie comptera pour 50% de la note et la qualité du livrable (code Python) pour 50%.

Le projet sera à présenter oralement le jour de l'examen.

LA PONDÉRATION DE L'ÉVALUATION DE L'UE À DES FINS DE DÉLIBÉRATION LORS DU CALCUL DE LA MOYENNE

Au sein du programme d'études, lors du calcul de la moyenne, la pondération de l'évaluation de l'unité d'enseignement est effectuée sur base des crédits affectés à cette unité d'enseignement et évalués

MODALITÉS D'ÉVALUATION PAR ACTIVITÉ D'APPRENTISSAGE

Activité d'apprentissage	Modalités d'évaluation
AAIN1005-A — Théorie	*À définir*
AAIN1005-B — Laboratoires	*À définir*

MODALITÉS CONCRÈTES D'ÉVALUATION

Une mention sur la fiche informera les étudiants du fait que les précisions sur les épreuves seront disponibles sur Moodle au plus tard 6 semaines avant l'évaluation.