

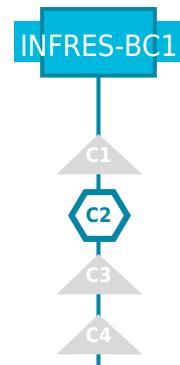
**Pourquoi cette UE ?**

Ce module associe DevOps, langage C et réseaux informatiques pour offrir une formation cohérente autour des fondamentaux du développement et de l'exploitation des systèmes. Le cours de C permet de comprendre la programmation bas niveau et les mécanismes proches de la machine ; le cours de réseaux apporte les bases de la communication entre systèmes ; le DevOps relie ces savoirs aux pratiques modernes d'intégration, d'automatisation et de déploiement. Leur complémentarité prépare l'élève à concevoir, gérer et optimiser des environnements informatiques complets.

**Eléments constitutifs de l'UE**

	coefficient	
INFRES_5_2-1 DevOps : fondamentaux	0	
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel	Nombre d'ECTS
84	42	4

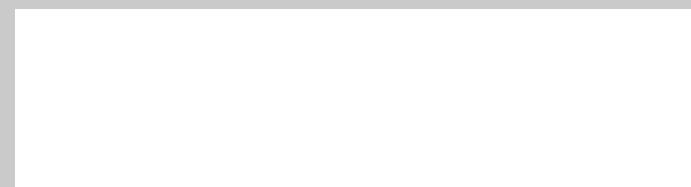
Alignement curriculaire

**Parmi les compétences visées par la formation, lesquelles sont développées dans cette UE ?**

- BC1** L'UE ne contribue pas à ce bloc de compétences
- BC1** L'UE contribue à ce bloc de compétences
- C1** Compétence non adressée dans cette UE
- C1** Compétence mise en œuvre dans cette UE
- C1** Compétence enseignée dans cette UE
- C1** Compétence évaluée dans cette UE
- C1** Compétence enseignée et évaluée dans cette UE

**Contexte et enjeux de l'enseignement**

Dans un monde numérique où les systèmes doivent évoluer rapidement tout en garantissant fiabilité et sécurité, l'approche DevOps s'impose comme une transformation majeure des pratiques informatiques. Elle favorise la collaboration entre développement et exploitation, en intégrant automatisation, supervision et déploiement continu. Ce cours sur les fondamentaux du DevOps s'inscrit dans ce contexte de mutation des méthodes de production logicielle. L'enjeu est d'offrir une vision d'ensemble des principes qui permettent d'accélérer la livraison de services, d'améliorer la qualité des applications et de renforcer la réactivité face aux besoins des utilisateurs. Comprendre ces bases est essentiel pour appréhender les évolutions des métiers du numérique et s'adapter à des environnements techniques en constante transformation.

**Prise en compte des dimensions socio-environnementales****Modalités d'enseignement et d'évaluation**

	Nb d'heures
Cours	
Cours intégré (cours + TD)	3
TD	
TP	
Projets	3
Travail en autonomie encadré	
Contrôles et soutenances	
Travail personnel	

**Prérequis**

Aucun

**Objectifs pédagogiques**

(à la fin de cet enseignement, l'étudiant sera capable de ...)

- Comprendre les concepts du DevOps
- Comprendre pourquoi on est arrivé au DevOps
- Quelles sont les grands principes mis en œuvre

**Activités**

(CM, TD, TP, projet, sortie terrain, etc. )

Conférence mêlant exposé théorique et démonstrations

**Évaluations et retours faits aux élèves**

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Pas d'évaluation

## Plan de cours

- Pourquoi fait-on de la qualité logicielle avec quelques cas emblématiques
- Quelles sont les différentes manières de faire du développement
- Qu'est ce qu'un build manager
- Le test
- L'automatisation et les plateformes de CI/CD
- Reporting et feedback

## Ressources et références

**Deprecated:** htmlspecialchars(): Passing null to parameter #1 (\$string) of type string is deprecated in **C:\Développement\syllabus\public\_html\views\syllabus\_template.php** on line **297**

**Contexte et enjeux de l'enseignement**

Le langage C occupe une place particulière dans l'histoire et la pratique de l'informatique. Utilisé pour le développement de systèmes d'exploitation, d'applications embarquées ou de logiciels à haute performance, il reste une référence incontournable pour comprendre les fondements de la programmation. Ce cours de programmation en C s'inscrit dans cette perspective : il offre à l'élève l'occasion de se confronter à un langage proche de la machine, exigeant rigueur et précision. L'enjeu est de renforcer la compréhension des mécanismes internes tels que la gestion de la mémoire, la compilation ou les structures de données, tout en développant de bonnes pratiques de programmation. Ce socle constitue une base solide pour aborder ensuite d'autres langages et paradigmes, mais aussi pour acquérir une culture informatique indispensable à tout futur ingénieur.

**Prise en compte des dimensions socio-environnementales****Modalités d'enseignement et d'évaluation****Prérequis**

Aucun

	Nb d'heures
Cours	
Cours intégré (cours + TD)	23
TD	
TP	
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	1
Travail personnel	20

**Objectifs pédagogiques**

(à la fin de cet enseignement, l'étudiant sera capable de ...)

- Maîtriser la syntaxe et les structures de base du langage C.
- Comprendre le cycle de compilation et d'exécution d'un programme.
- Savoir utiliser les types, pointeurs, tableaux et structures.
- Gérer la mémoire de manière explicite et efficace.
- Appliquer des bonnes pratiques de programmation et de débogage.

**Activités**

(CM, TD, TP, projet, sortie terrain, etc. )

Cours et TD non séparés. La participation des étudiants est sollicitée tout au long de la séance.

**Évaluations et retours faits aux élèves**

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Evaluation : 2 contrôles surveillés d'1heure, même coefficient.

Retour sur l'évaluation fait à l'élève : Copies corrigées consultables sur demande

## Plan de cours

Le cours est divisé en 2 parties. La première partie consiste à aborder les notions de base de la programmation procédurale et des structures de données de type simple, tableaux et enregistrement indépendamment d'un langage :

- Chapitre 1 : Structure d'un programme - les types de données (entier, réel, caractère, booléen) et les structures de contrôle (séquentielle, sélective, répétitive).
- Chapitre 2 : Les tableaux statiques - méthodes de tri.

La deuxième partie de ce cours permet à des étudiants ne possédant aucune notion dans un langage informatique d'apprendre à développer des petits programmes en langage C.

- Chapitre 3 : Présentation de l'histoire de l'informatique, des origines à nos jours.
- Chapitre 4 : Les bases de l'informatique, notamment les différents types de données utilisées ainsi que la notion d'algorithme.
- Chapitre 5 : Les différentes instructions du langage C.
- Chapitre 6 : Les structures de contrôles.
- Chapitre 7 : Les fonctions.

## Ressources et références

Cours et corrections déposés sur Campus2.

**Contexte et enjeux de l'enseignement**

Pour s'assurer un déploiement efficace, sécurisé et évolutif d'un réseau informatique, sa mise en place devrait suivre les différentes phases suivantes: • Analyse des besoins et planification ; • Conception de l'architecture réseau ; • Installation et configuration des équipements réseaux ; • Test et validation du réseau ; • Déploiement et mise en production. La phase de test et validation est très souvent négligée, par manque de temps ou de budget. Elle est pourtant très importante, car elle réduit les probabilités de déploiement d'un réseau dysfonctionnel (et non conforme aux besoins). Elle permet aussi de tester plusieurs solutions logicielles et matérielles à moindres coûts. Cet enseignement vise à la maîtrise d'un outil de virtualisation nommé GNS3. Cet outil permet de déployer un réseau virtuel constitué d'équipements réseaux usuels. Il a la particularité d'être relativement simple d'utilisation, de disposer d'une communauté d'utilisateur et de concepteur importante, d'être gratuit et de pouvoir virtualiser de nombreux équipements réseaux couramment utilisés. Il est l'outil idéal pour une phase de test et de validation d'une architecture réseau.

**Prise en compte des dimensions socio-environnementales**

ODD9 - Industrie, innovation et infrastructure

**Prérequis**

Maîtrise des protocoles réseaux de base (adressage et routage IPv4, VLAN...) et de la virtualisation

**Modalités d'enseignement et d'évaluation**

	Nb d'heures
Cours	
Cours intégré (cours + TD)	23
TD	
TP	
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	4
Travail personnel	12

**Objectifs pédagogiques**

(à la fin de cet enseignement, l'étudiant sera capable de ...)

- Installer l'outil GNS3
- Ajouter et modifier des modèles de nœud (équipements réseaux, serveurs, clients...)
- Mettre en place d'un réseau virtuel complexe (VLAN, routage IPv4 et IPv6, multicast, tunnel GRE...)
- Tester le réseau virtuel, et rendre compte du test

**Activités**

(CM, TD, TP, projet, sortie terrain, etc. )

TP d'installation, de création de modèle de nœud, de mise en place de réseau virtuel

**Évaluations et retours faits aux élèves**

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

TP avec rapports à déposer sur Campus, et tests de fonctionnement en direct avec le formateur

**Plan de cours**

- Installation et configuration de GNS3
- Création et modification de modèle de noeuds (VyOS, Alpine Linux, OpenvSwitch)
- Principes de base d'IPv6 et du multicast
- Mise en place d'un réseau avec VLAN, et d'un réseau multicast

**Ressources et références**

Documentation GNS3 en ligne, machine virtuelle GNS3

Documentation interne

Cours Multicast et IPv6

**Contexte et enjeux de l'enseignement**

La plupart des applications informatiques interagissent constamment avec des réseaux, que ce soit via des API, des bases de données ou le cloud. Une bonne maîtrise des concepts réseau permet d'optimiser les échanges, d'améliorer la performance et de renforcer la sécurité des applications en prévenant les vulnérabilités. Comprendre le réseau facilite aussi la résolution des problèmes et le travail avec les équipes DevOps. Dans un contexte cloud ou microservices, cette compétence devient essentielle pour gérer la connectivité et optimiser l'infrastructure. Elle ouvre aussi des perspectives vers des rôles comme architecte cloud ou ingénieur DevOps. Enfin, dans des domaines comme les jeux en ligne ou le streaming, savoir gérer la latence et la bande passante améliore l'expérience utilisateur. Sans devenir administrateur réseau, un développeur gagne en efficacité et en polyvalence en maîtrisant ces notions.

**Prise en compte des dimensions socio-environnementales**

ODD9 - Industrie, innovation et infrastructure

**Prérequis**

Aucun

**Modalités d'enseignement et d'évaluation**

	Nb d'heures
Cours	
Cours intégré (cours + TD)	23
TD	
TP	
Projets	
Travail en autonomie encadré	
Contrôles et soutenances	4
Travail personnel	10

**Objectifs pédagogiques**

(à la fin de cet enseignement, l'étudiant sera capable de ...)

**Activités**

(CM, TD, TP, projet, sortie terrain, etc.)

- Identifier les différents équipements réseaux et outils associés
- Analyser une problématique réseau et en déduire une organisation, une topologie, les techniques et équipements à planter, un plan d'adressage.
- Paramétriser les principaux équipements réseaux et terminaux
- Faire des choix de conception d'application en fonction des besoins de communication
- Dialoguer avec les experts et administrateurs réseaux

10 heures de cours magistral et 10 heures de travaux pratiques.  
Les TP sont basés sur le simulateur de réseau Packet Tracer.

**Évaluations et retours faits aux élèves**

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

- 1 QCM de 1 heure 15 mn, avec certification Cisco (CCNA Présentation des réseaux).
- 1 TP de mise en place d'un réseau local.

Retour : Copies corrigées consultables sur demande

## Plan de cours

- Définition d'un réseau de communication, des services réseaux, des usagers et des informations ;
- Le réseau en mode paquet, et le modèle TCP/IP ;
- Les équipements réseaux (routeur, commutateur, serveur DNS...), les protocoles (commutation, VLAN...) et l'architecture réseau ;
- L'adressage et le routage IPv4 ;
- Les services réseaux essentiels (DHCP, DNS, NAT)
- Mise en place d'un petit réseau informatique

## Ressources et références

Cours et corrections déposés sur Campus2.

Simulateur de réseau Packet Tracer