

PLAN D'ÉTUDES

Informatique et systèmes de communication (ISC)

2025–2026

Secrétariat académique
Responsable de filière (RF)

Éliane Favre
Dr Pierre-André Mudry

Arrêté le 10 septembre 2025
Version définitive

cdee5a

Table des matières

Préambule

1.1	Introduction au plan d'études	5
1.2	Règlements de la filière	5
1.3	Programme de formation (à plein temps)	7
1.4	Programme de formation (à temps partiel)	9
1.5	Calendrier académique	11

Descriptif des modules

2.6	100 - Sciences IT 1	13
2.7	101 - Fondements de la programmation 1	21
2.8	102 - Architecture matérielle	25
2.9	103 - Sécurité et communication	29
2.10	104 - Humanités 1	33
2.11	105 - Summer school 1	39
2.12	200 - Sciences IT 2	41
2.13	201 - Fondements de la programmation 2	45
2.14	202 - Systèmes et concurrence	53
2.15	203 - Humanités 2	59
2.16	204 - Sciences IT 3	65
2.17	205 - Advanced programming	71
2.18	206 - Text and vision	79
2.19	207 - Summer school 2	85
2.20	301 - Machine learning	87
2.21	302 - Data computation	91
2.22	303 - Semester project	95
2.23	304 - Computer engineering option	97
2.24	305 - Data science option	105
2.25	330 - Bachelor thesis	115

1.1. Introduction au plan d'études

Ce document décrit le plan d'études de la formation bachelor en *Informatique et Systèmes de communication* auprès de la *Haute Ecole d'Ingénierie* (ci-après *HEI*) de la *HES-SO Valais-Wallis*.

La formation ISC est constituée de différents modules de formation qui donnent accès à un certain nombre de crédits ECTS (*European Credit Transfer System*), tels que définis dans le modèle de Bologne. La formation bachelor HES en *Informatique et systèmes de communication* comprend un total de 180 crédits répartis à raison de 60 crédits par année d'étude (dans le modèle à plein temps). Les crédits ECTS visent notamment à faciliter la mobilité étudiante entre les établissements durant le cursus de formation.

Le plan d'études est composé de plusieurs éléments :

- des informations sur les différents **règlements** de la filière ISC;
- d'un **programme de formation à plein temps**, qui présente de manière synthétique la formation avec les modules qui la composent durant les trois années que dure la formation;
- d'un **programme de formation à temps partiel**, qui présente de manière synthétique la formation avec les modules qui la composent durant les quatre années que dure la formation à temps partiel;
- des **fiches de module**, qui décrivent les contenus, objectifs et compétences développés durant la durée des études dans leur forme modulaire. Ces fiches de modules stipulent également les différentes personnes responsables des modules et des unités de cours ainsi que les différentes règles de calcul de notes, les modalités de répétition et d'éventuelles remédiations lorsque celles-ci sont possibles.
- du **calendrier académique**, qui présente les dates importantes pour l'année académique en cours.

1.2. Règlements de la filière

On retrouvera toutes les informations relatives au règlement de la filière (règles de promotion, échec, ...) sur les liens suivants :

- **Règlement des études** – Le règlement des études de la Haute Ecole d'Ingénierie [disponible en ligne](#) fait foi. Il est à considérer en supplément du [règlement du domaine IA](#);
- **Calcul des notes des modules** – [Directive](#) relative à l'évaluation et à la validation des modules des filières bachelor de la Haute Ecole d'Ingénierie.

En cas de doutes, n'hésitez pas à prendre contact avec le secrétariat de la filière par mail (info.isc@hevs.ch) ou par téléphone au ☎ +41 58 606 93 60.

Programme de formation à temps plein – Filière ISC

Année académique 2025–2026

	Nom du module	ECTS	Unités d'enseignement Cours	Poids	Périodes automne	Périodes printemps
1ère année, 60 ECTS	100 – Sciences IT 1	20	100.1 – Algèbre linéaire 1	3	4	
			100.2 – Algèbre linéaire 2	4		4
			100.3 – Analyse 1	5	6	
			100.4 – Analyse 2	4		4
			100.5 – Mathématiques discrètes	4		4
			100.8 – Complément sciences IT (opt.)		(2)	(2)
	101 – Fondements de la programmation 1	12	101.1 – Programmation impérative	6	6	
			101.2 – Programmation orientée-objets	6		6
			101.8 – Complément fondements prog. 1 (opt.)		(2)	(2)
	102 – Architecture matérielle	9	102.1 – Systèmes numériques	5	6	
			102.2 – Architecture des ordinateurs	4		4
			102.8 – Complément systèmes numériques (opt.)		(2)	
2ème année, 60 ECTS	103 – Sécurité et communication	6	103.1 – Réseaux IP	3	4	
			103.2 – Cryptographie et sécurité	3		4
	104 – Humanités 1	9	104.1 – Anglais 1	4	4	
			104.2 – Anglais 2	2		2
			104.4 – Communication et expression	3		3
	105 – Summer school 1	4	105.1 – Data challenge summer school	4	Summer school 1	
	200 – Sciences IT 2	6	200.1 – Computational physics 1	3	4	
			200.2 – Probabilités et statistiques IT	3	4	
	201 – Fondements de la programmation 2	12	201.1 – Algorithmes et structures de données	4	4	
			201.2 – Génie logiciel	4	4	
			201.3 – Bases de données relationnelles	4	4	
3ème année, 60 ECTS	202 – Systèmes et concurrence	6	202.1 – Systèmes d'exploitation	4	4	
			202.2 – Programmation concurrente	2	2	
	203 – Humanités 2	4	203.1 – Éthique et droit IT	2	2	
			203.3 – Engineering in computer science	2	3	
	204 – Sciences IT 3	7	204.1 – Signal and information	4		4
			204.2 – Computational physics 2	3		4
	205 – Advanced programming	11	205.1 – Functional programming	3		3
			205.2 – Beyond relational databases	4		4
			205.3 – Full-stack Web development	4		4
	206 – Text and vision	11	206.1 – Data processing and visualization	4		4
			206.2 – Model driven machine vision	4		4
			206.3 – Formal and natural languages	3		3
3ème année, 60 ECTS	207 – Summer school 2	3	207.1 – Mobile development summer school	3	Summer school 2	
	301 – Machine learning and AI	13	301.1 – Optimization, ML and deep learning	13	Bloc	
	302 – Data computation	13	302.1 – Big data, cloud, infras & HPC	14	Bloc	
	303 – Semester project	4	303.1 – Hands-on development project	3	Bloc	
	304 – Major option track	10	304.1 – Robot control and generative AI	10	} A choix	
			304.2 – Medical informatics and data science	10		
			304.3 – Modern compiler construction	10		
			304.4 – LLMs and agents	10		
	305 – Minor option track	5	305.1 – IoT and distributed AI	5	} A choix	
			305.2 – Applied cybersecurity : defense and attack	5		
			305.3 – Real-world data science	5		
			305.4 – Computer graphics and games	5		
			305.5 – Reverse engineering	5		
3ème année, 60 ECTS	330 – Bachelor thesis	15	305.4 – Digital energy management	5		
			330.1 – Travail de bachelor	15	Bloc	

Programme de formation à temps partiel – Filière ISC

Année académique 2025–2026

	Nom du module	ECTS	Unités d'enseignement Cours	Poids	Périodes automne	Périodes printemps
1ère année, temps partiel 42 ECTS	100 – Sciences IT 1	20	100.1 – Algèbre linéaire 1	3	4	
			100.2 – Algèbre linéaire 2	4		4
			100.3 – Analyse 1	5	6	
			100.4 – Analyse 2	4		4
			100.5 – Mathématiques discrètes	4		4
			100.8 – Complément sciences IT (opt.)		(2)	(2)
	101 – Fondements de la programmation 1	12	101.1 – Programmation impérative	6	6	
			101.2 – Programmation orientée-objets	6		6
			101.8 – Complément fondements prog. 1 (opt.)		(2)	(2)
	103 – Sécurité et communication	6	103.1 – Réseaux IP	3	4	
			103.2 – Cryptographie et sécurité	3		4
105 – Summer school 1	4	105.1 – Data challenge summer school	4	Summer school 1		
2ème année, temps partiel 41 ECTS	102 – Architecture matérielle	9	102.1 – Systèmes numériques	5	6	
			102.2 – Architecture des ordinateurs	4		4
			102.8 – Complément systèmes numériques (opt.)		(2)	
	104 – Humanités 1	9	104.1 – Anglais 1	4	4	
			104.2 – Anglais 2	2		2
			104.4 – Communication et expression	3		3
	201 – Fondements de la programmation 2	12	201.1 – Algorithmes et structures de données	4	4	
			201.2 – Génie logiciel	4	4	
			201.3 – Bases de données relationnelles	4	4	
	205 – Advanced programming	11	205.1 – Functional programming	3		3
			205.2 – Beyond relational databases	4		4
205.3 – Full-stack Web development			4		4	
3ème année, temps partiel 37 ECTS	200 – Sciences IT 2	6	200.1 – Computational physics 1	3	4	
			200.2 – Probabilités et statistiques IT	3	4	
	202 – Systèmes et concurrence	6	202.1 – Systèmes d'exploitation	4	4	
			202.2 – Programmation concurrente	2	2	
	203 – Humanités 2	4	203.1 – Éthique et droit IT	2	2	
			203.3 – Engineering in computer science	2	3	
	204 – Sciences IT 3	7	204.1 – Signal and information	4		4
			204.2 – Computational physics 2	3		4
	206 – Text and vision	11	206.1 – Data processing and visualization	4		4
			206.2 – Model driven machine vision	4		4
			206.3 – Formal and natural languages	3		3
207 – Summer school 2	3	207.1 – Mobile development summer school	3	Summer school 2		
4ème année, temps plein 60 ECTS	301 – Machine learning and AI	13	301.1 – Optimization, ML and deep learning	13	Bloc	
	302 – Data computation	13	302.1 – Big data, cloud, infras & HPC	14	Bloc	
	303 – Semester project	4	303.1 – Hands-on development project	3	Bloc	
	304 – Major option track	10	304.1 – Robot control and generative AI	10	} A choix	
			304.2 – Medical informatics and data science	10		
			304.3 – Modern compiler construction	10		
			304.4 – LLMS and agents	10		
	305 – Minor option track	5	305.1 – IoT and distributed AI	5	} A choix	
			305.2 – Applied cybersecurity : defense and attack	5		
			305.3 – Real-world data science	5		
			305.4 – Computer graphics and games	5		
305.5 – Reverse engineering			5			
305.4 – Digital energy management			5			
330 – Bachelor thesis	15	330.1 – Travail de bachelor	15	Bloc		

Calendrier académique ISC - 2025/26

Semaine	du	au	Bachelor of Science en Informatique et systèmes de communication (ISC)							
			1ère année	2ème année	3ème année	Fêtes et congés	Sem. acad.			
38	15.09.2025	19.09.2025					1			
39	22.09.2025	26.09.2025					2			
40	29.09.2025	03.10.2025					3			
41	06.10.2025	10.10.2025					4			
42	13.10.2025	17.10.2025					5			
43	20.10.2025	24.10.2025					6			
44	27.10.2025	31.10.2025	Travail autonome, congés		Travail de semestre		- 7			
45	03.11.2025	07.11.2025					7 8			
46	10.11.2025	14.11.2025					8 9			
47	17.11.2025	21.11.2025					9 10			
48	24.11.2025	28.11.2025					10 11			
49	01.12.2025	05.12.2025					11 12			
50	08.12.2025	12.12.2025				Immaculée conception 8.12	12 13			
51	15.12.2025	19.12.2025					13 14			
52	22.12.2025	26.12.2025					Vacances de Noël			14 15 15 -
1	29.12.2025	02.01.2026							Voyage d'études	
2	05.01.2026	09.01.2026							Vacances intersemestre	
3	12.01.2026	16.01.2026					Préparation des examens, congés			
4	19.01.2026	23.01.2026	Examens de module							
5	26.01.2026	30.01.2026	Vacances intersemestre							
6	02.02.2026	06.02.2026					1			
7	09.02.2026	13.02.2026						2		
8	16.02.2026	20.02.2026						3		
9	23.02.2026	27.02.2026						4		
10	02.03.2026	06.03.2026						5		
11	09.03.2026	13.03.2026						6		
12	16.03.2026	20.03.2026						7		
13	23.03.2026	27.03.2026				Saint-Joseph 19.3				
14	30.03.2026	03.04.2026				Vendredi Saint 3.4				
15	06.04.2026	10.04.2026	Vacances de Pâques				8			
16	13.04.2026	17.04.2026						9		
17	20.04.2026	24.04.2026						10		
18	27.04.2026	01.05.2026						11		
19	04.05.2026	08.05.2026						11'		
20	11.05.2026	15.05.2026						Cours du mardi remplacés par les cours du vendredi		Ascension 14.5 et 15.5
21	18.05.2026	22.05.2026			Travail de bachelor		13			
22	25.05.2026	29.05.2026				Pentecôte 25.5	14			
23	01.06.2026	05.06.2026				Fête-Dieu 4.6	15			
24	08.06.2026	12.06.2026								
25	15.06.2026	19.06.2026				Préparation des examens, congés				
26	22.06.2026	26.06.2026	Examens de module							
27	29.06.2026	03.07.2026								
28	06.07.2026	10.07.2026								
29	13.07.2026	17.07.2026								
30	20.07.2026	24.07.2026	Vacances d'été							
31	27.07.2026	31.07.2026								
32	03.08.2026	07.08.2026								
33	10.08.2026	14.08.2026								
34	17.08.2026	21.08.2026	Remédiations	Remédiations			Défense TB	Remédiations 1ère et 2ème année		
35	24.08.2026	28.08.2026	Summer school 1	Summer school 2					21.8 : expo des TBs, 24.8 : Monthey	
36	31.08.2026	04.09.2026								
37	07.09.2026	11.09.2026	Semaine d'interruption							
38	14.09.2026	18.09.2026	Reprise des cours année académique 2026-2027							

	Libre
	En formation

Toute modification ultérieure demeure réservée

100 – Sciences IT 1

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Sciences de l'ingénierie	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Sciences IT 1	Code du module	100
Type	Obligatoire	Crédits ECTS	20
Langue	Français	Année académique	Première année
Responsable(s)	Dr Matthieu Jacquemet	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
100.1	ALin1 Algèbre linéaire 1	4							
100.2	ALin2 Algèbre linéaire 2		4						
100.3	Ana1 Analyse 1	6							
100.4	Ana2 Analyse 2		4						
100.5	MDis Mathématiques discrètes		4						

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	260 h	240 h	500 h

2. Description du module

Ce module a pour but de fournir des bases mathématiques nécessaires à l'ingénierie en général et à l'*Informatique et systèmes de communication* en particulier. Plus précisément, le cours *Analyse* traite des fonctions et du calcul différentiel et intégral, le cours *Algèbre linéaire* aborde tout d'abord la géométrie vectorielle en dimension arbitraire pour déboucher sur le calcul matriciel et les applications linéaires au 2e semestre. Finalement, le cours *Mathématiques discrètes* propose une introduction aux structures discrètes (ensembles, graphes, combinatoire) et à différents types de logiques et de preuves.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☐ Avoir suivi le(s) module(s)
- ☒ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

- connaître les concepts et techniques de base de l'analyse (calcul algébrique, (in)équations, fonctions, limites, calcul différentiel et intégral) **(C)**
- connaître les concepts et techniques de base de l'algèbre linéaire (géométrie analytique et vectorielle, trigonométrie, systèmes d'équations linéaires, algèbre des matrices, diagonalisation) **(C)**
- connaître les concepts et techniques de base des mathématiques discrètes (théorie des ensembles, des graphes et des logiques propositionnelles et des prédicats) **(C)**
- illustrer ces concepts et techniques à l'aide d'exemples et de contre-exemples **(C)**
- utiliser ces concepts et techniques en vue de résoudre des problèmes **(A)**
- interpréter géométriquement des propriétés algébriques et vice-versa **(A)**
- manipuler des expressions algébriques et exécuter des procédures de résolution de manière structurée et sans faute **(A)**
- utiliser les théorèmes existants pour prouver d'autres énoncés **(J)**
- communiquer son raisonnement de manière claire, structurée et complète **(J)**

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{3 \cdot m_{\text{ALin1}} + 4 \cdot m_{\text{ALin2}} + 5 \cdot m_{\text{Ana1}} + 4 \cdot m_{\text{Ana2}} + 4 \cdot m_{\text{MDis}}}{20}$$

avec :

- m_{ALin1} – moyenne des notes en *Algèbre linéaire 1*;
- m_{ALin2} – moyenne des notes en *Algèbre linéaire 2*;
- m_{Ana1} – moyenne des notes en *Analyse 1*;
- m_{Ana2} – moyenne des notes en *Analyse 2*;
- m_{MDis} – moyenne des notes en *Mathématiques discrètes*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

En cas de mise à niveau après le 1er semestre, le module est considéré **comme un échec**.

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant-e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 100.1 – Algèbre linéaire 1 (ALin1)

Périodes/sem	4	Semestre	Automne
Enseignant-e	Enseignant-e tronc commun maths	Coefficient	3

Description courte / objectifs

- Connaître et appliquer les concepts et techniques de base de la géométrie analytique et vectorielle en dimensions 2 et 3, de la trigonométrie, et des systèmes d'équations linéaires
- Illustrer ces concepts et techniques à l'aide d'exemples et de contre-exemples
- Utiliser ces concepts et techniques en vue de résoudre des problèmes
- Communiquer son raisonnement de manière claire, structurée et complète

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Géométrie analytique du plan** : équations de droite et de cercle, angles de deux droites, intersections
- **Trigonométrie** : cercle trigonométrique, fonctions trigonométriques, équations trigonométriques, résolution de triangles
- **Géométrie vectorielle de l'espace à 3 dimensions** : combinaison linéaire, bases, produit scalaire, équations de droite, de plan, de sphère, produit vectoriel
- **Systèmes d'équations linéaires** : algorithme de Gauss, rang, ensembles de solutions, systèmes homogènes

Support de cours

Cours, formulaire, séries d'exercices.

Particularité d'organisation

Ce cours fait partie du tronc commun partagé entre toutes les filières de la Haute Ecole d'Ingénierie. Un cours de complément (2 périodes par semaine) est offert à toutes les personnes souhaitant ou ayant besoin d'approfondir leurs connaissances.

Unité d'enseignement 100.2 – Algèbre linéaire 2 (ALin2)

Périodes/sem	4	Semestre	Printemps
Enseignant-e	Dr Matthieu Jacquemet	Coefficient	4

Description courte / objectifs

- Connaître et appliquer les concepts et techniques de base de la géométrie analytique et vectorielle en dimensions quelconques, du calcul matriciel, des applications linéaires et de la diagonalisation de matrices
- Illustrer ces concepts et techniques à l'aide d'exemples et de contre-exemples
- Utiliser ces concepts et techniques en vue de résoudre des problèmes
- Communiquer son raisonnement de manière claire, structurée et complète

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Géométrie vectorielle dans l'espace à n dimensions** : norme, produit scalaire, équations de droite, d'hyperplan, d'hypersphère
- **Calcul matriciel** : déterminant, inverse, algorithme de Gauss-Jordan, forme matricielle des systèmes d'équations linéaires
- **Applications linéaires** : quelques transformations géométriques (dilatation, projection, symétrie, rotation), matrice d'une application linéaire, changements de base, matrice de passage
- **Diagonalisation** : valeurs propres, vecteurs propres, espaces propres, diagonalisation
- **Application du calcul matriciel** : matrices orthogonales, matrices stochastiques

Support de cours

Cours, formulaire, séries d'exercices.

Outils utilisés

Langage de programmation [Python](#).

Bibliographie

- Peter Farrell, *Math Adventures with Python*, ISBN-13 978-1593278670, 2019.

Unité d'enseignement 100.3 – Analyse 1 (Ana1)

Périodes/sem	6	Semestre	Automne
Enseignant·e	Enseignant·e tronc commun maths	Coefficient	5

Description courte / objectifs

- Connaître et appliquer les concepts et techniques de base de l'algèbre, connaître les fonctions usuelles et leurs propriétés, connaître les concepts et techniques de base du calcul des limites
- Illustrer ces concepts et techniques à l'aide d'exemples et de contre-exemples
- Utiliser ces concepts et techniques en vue de résoudre des problèmes
- Communiquer son raisonnement de manière claire, structurée et complète

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Manipulations algébriques de base** : types de nombres, les opérations et leurs propriétés, puissances et racines, exponentielles et logarithmes
- **Polynômes** : trinôme du second degré, équation quadratique, divisibilité par $(x - a)$, schéma de Horner, factorisation de polynômes
- **Equations et inéquations**
- **Suites et sommes** : suites de nombres, sommes, binôme de Newton et coefficient binomial
- **Fonctions** : définitions, fonction inverse, symétries du graphe, limites et asymptotes

Support de cours

Cours, formulaire, séries d'exercices.

Particularité d'organisation

Ce cours fait partie du tronc commun partagé entre toutes les filières de la Haute Ecole d'Ingénierie. Un cours de complément (2 périodes par semaine) est offert à toutes les personnes souhaitant ou ayant besoin d'approfondir leurs connaissances.

Unité d'enseignement 100.4 – Analyse 2 (Ana2)

Périodes/sem	4	Semestre	Printemps
Enseignant-e	Enseignant-e tronc commun maths	Coefficient	4

Description courte / objectifs

- Connaître les concepts et techniques de base du calcul différentiel et du calcul intégral
- Illustrer ces concepts et techniques à l'aide d'exemples et de contre-exemples
- Utiliser ces concepts et techniques en vue de résoudre des problèmes
- Communiquer son raisonnement de manière claire, structurée et complète

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Dérivées** : définitions, règles de dérivation
- **Applications des dérivées** : croissance et extrema, optimisation, théorèmes de L'Hospital et de Taylor, convexité et point d'inflexion
- **Intégrales** : intégrale définie et aire sous une courbe, intégrale indéfinie et primitives, théorème fondamental du calcul différentiel et intégral
- **Méthodes d'intégration** : intégration par partie, substitution et changement de variable, décomposition en fractions simples

Support de cours

Cours, formulaire, séries d'exercices.

Particularité d'organisation

Ce cours fait partie du tronc commun partagé entre plusieurs filières de la HEI. Un cours de complément (2 périodes par semaine) est offert à toutes les personnes souhaitant ou ayant besoin d'approfondir leurs connaissances.

Unité d'enseignement 100.5 – Mathématiques discrètes (MDi s)

Périodes/sem	4	Semestre	Printemps
Enseignant-e	Dr Matthieu Jacquemet	Coefficient	4

Description courte / objectifs

- Connaître les concepts et techniques de base de la théorie des ensembles, des graphes et des logiques propositionnelle et des prédicats
- Illustrer ces concepts et techniques à l'aide d'exemples et de contre-exemples
- Utiliser ces concepts et techniques en vue de résoudre des problèmes
- Utiliser les théorèmes existants pour prouver d'autres énoncés
- Communiquer son raisonnement de manière claire, structurée et complète

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Différents types de preuve** : contraposition, induction, directe
- **Théorie des ensembles** : intersection, union, complément, différence
- **Combinatoire** : dénombrement, coefficient binomial et multinomial, inclusion-exclusion
- **Théorie des graphes** : isomorphisme de graphes, composantes, incidence et adjacence, graphes eulériens, arbres, algorithme de Kruskal, graphes planaires, matching et théorème de Hall
- **Logique propositionnelle** : syntaxe et sémantique, tables de vérité, équivalence, tautologie, théories de preuves
- **Logique des prédicats** : éléments de logique des prédicats, variables libres et liées, langages de premier ordre

Support de cours

Cours, formulaire, séries d'exercices.

101 – Fondements de la programmation 1

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Base des TIC	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Fondements de la programmation 1	Code du module	101
Type	Obligatoire	Crédits ECTS	12
Langue	Français	Année académique	Première année
Responsable(s)	Dr Pierre-André Mudry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
101.1	PImp Programmation impérative	6							
101.2	PObj Programmation orientée-objets		6						

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	140 h	160 h	300 h

2. Description du module

Créer du code de qualité, lisible, fiable et que l'on peut maintenir sont des besoins centraux pour tous les projets logiciels. Dans ce module, les étudiant·e·s construisent les fondations dans ce sens en découvrant deux facettes essentielles du développement : la *programmation impérative* et la *programmation orientée-objets*. Par des projets concrets et du travail en équipe sont également posées les bases du développement professionnel en équipe avec une perspective d'ingénieur·e·.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☐ Avoir suivi le(s) module(s)
- ☒ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

- identifier les éléments syntaxiques et sémantiques d'un programme (**J**)
- formuler un problème de traitement de données en termes d'algorithmes impératifs (**C**)
- sélectionner les types adéquats et les opérations afin de réaliser des algorithmes (**C**)

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

- choisir les structures de données appropriées pour résoudre des problèmes **(A)**
- réaliser des applications simples de différents types (graphiques, jeux) en groupe **(A)**
- reproduire le cycle de développement d'un programme en groupe à l'aide des outils idoines, notamment de gestion de version ([git](#)) et de tests unitaires **(C)**

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{6 \cdot m_{\text{PImp}} + 6 \cdot m_{\text{PObj}}}{12}$$

avec :

- m_{PImp} – moyenne des notes en *Programmation impérative*;
- m_{PObj} – moyenne des notes en *Programmation orientée-objets*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant-e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 101.1 – Programmation impérative (PImp)

Périodes/sem	6	Semestre	Automne
Enseignant·e·s	Dr P.-A. Mudry, Dr Renaud Richardet	Coefficient	6

Description courte / objectifs

Dans ce cours sont posées les bases de la programmation en utilisant un style dit *impératif* – qui agit en modifiant l'état de la mémoire. Des exercices et des projets permettront de maîtriser les principes fondamentaux de la programmation et du code tout en développant également la capacité à penser et à résoudre les problèmes comme un·e programmeur·se.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- Etat** : variables et valeurs, assignation, types de base et conversions, opérateurs et expressions
- Flux de contrôle** : séquence d'instructions, instructions conditionnelles, répétitions
- Fonctions** : passage de paramètres, valeurs de retour, récursion simple
- Manipulation de chaînes de caractères** : `String`, méthodes et interpolation
- Programming in practice** : outils de *debug* et de travail en groupe (`git`), flux de données, grammaires, `terminal`
- Types structurés** : `Option`, `Null` et `Unit`, tableaux, tuples et types de données complexes

Support de cours

Slides, vidéos, exercices en ligne, laboratoires et mini-projets.

Outils utilisés

- Langage de programmation `Scala 2.13` ou `3.X`
- Environnement de développement `IntelliJ`

Bibliographie

- [Scala reference manual](#)
- Martin Odersky *et al.*, *Programming in Scala, Fifth edition*, ISBN-13 978-0997148008, 2021.

Particularité d'organisation

Un cours de complément (2 périodes par semaine) est offert à toutes les personnes souhaitant ou ayant besoin d'approfondir leurs connaissances.

Unité d'enseignement 101.2 – Programmation orientée-objets (PObj)

Périodes/sem	6	Semestre	Printemps
Enseignant·e·s	Dr P.-A. Mudry, Dr Renaud Richardet	Coefficient	6

Description courte / objectifs

À partir des bases acquises dans le cours *101.1 Programmation impérative*, de nouvelles compétences en programmation orientée-objets sont développées. Grâce à cela, il deviendra possible de construire de nouvelles abstractions encapsulant données et méthodes. Des structures de données plus complexes ([List](#), [Map](#), ...) seront également introduites tout comme des notions de typages plus avancées, notamment à l'aide de la généricité et de la réflexion. La mise en pratique sera effectuée par le biais de laboratoires appliqués ainsi que sur un projet plus conséquent de jeu vidéo réalisé en groupe.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre (poids précisé en début de semestre). Cette unité de cours n'a pas d'examen de semestre.

Contenu (mots-clefs)

1. **Orienté-objet** : classes, objets et propriétés, héritage, interfaces et délégation, classes données
2. **Typage** : généricité et polymorphisme, égalité structurelle et référentielle, structures de données complexes et abstraites ([List](#), [Map](#)), sérialisation, réflexion (si le temps le permet)
3. **Éléments d'algorithmique** : tris et éléments de la théorie de la complexité, récursion et structures de données récursives (listes chaînées)

Support de cours

Slides, vidéos, exercices en ligne, laboratoires et mini-projets.

Outils utilisés

- Langage de programmation [Scala 2.13 ou 3.X](#)
- Environnement de développement [IntelliJ](#)

Bibliographie

- [Scala reference manual](#)
- Martin Odersky *et al.*, *Programming in Scala, Fifth edition*, ISBN-13 978-0997148008, 2021

Particularité d'organisation

Un cours de complément (2 périodes par semaine) est offert à toutes les personnes souhaitant ou ayant besoin d'approfondir leurs connaissances.

102 – Architecture matérielle

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Base des TIC	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Architecture matérielle	Code du module	102
Type	Obligatoire	Crédits ECTS	9
Langue	Français	Année académique	Première année
Responsable(s)	Silvan Zahno	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
102.1	DiD	Systèmes numériques	6						
102.2	ArchOrd	Architecture des ordinateurs	4						

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	120 h	110 h	230 h

2. Description du module

Le monde digital est celui composé par des 0 et des 1. Dès lors, comment à partir de ces deux nombres est-il possible de réaliser des calculs ? Comment fait un processeur pour exécuter des instructions ? Quels sont les éléments logiques qui le composent et comment sont-ils agencés ? Quel est donc le rapport entre les bits et la logique digitale ?

Ces quelques questions illustrent les contenus abordés dans ce module dans lequel vous apprendrez à structurer les portes logiques et les éléments de mémoire pour réaliser, au final, un processeur complet.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☐ Avoir suivi le(s) module(s)
- ☒ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- interpréter le cahier des charges d'un système matériel simple et d'en réaliser les fonctions logiques qui en découlent selon les principes de base de la conception et les méthodologies proposées (**A**)

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

- représenter et expliquer la structure interne d'un processeur simple **(C)**
- effectuer des calculs de performance des processeurs et les comparer entre eux **(C)**
- sélectionner les blocs matériels de calcul adéquats pour réaliser une fonction spécifique **(A)**
- mettre en pratique le cycle de développement hardware **(C)**

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{5 \cdot m_{\text{DiD}} + 4 \cdot m_{\text{ArchOrd}}}{9}$$

avec :

- m_{DiD} – moyenne des notes en *Systèmes numériques*;
- m_{ArchOrd} – moyenne des notes en *Architecture des ordinateurs*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 102.1 – Systèmes numériques (DiD)

Périodes/sem	6	Semestre	Automne
Enseignant·e·s	Silvan Zahno, Dr Andrea Guerrieri	Coefficient	5

Description courte / objectifs

Ce cours a pour objectifs de :

- poser les bases de la logique numérique ainsi que les éléments constitutifs individuels des circuits numériques. Les cours en classe sont complétés par des exercices, des laboratoires et des projets
- développer la capacité à concevoir et à réaliser des circuits numériques simples de manière autonome à l'aide d'outils EDA (*Electronic design automation*) et de simulation
- développer les méthodologies de l'ingénieur·e en divisant un grand problème en plusieurs petits

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Un examen intermédiaire ainsi que la notation d'un projet (moitié de la note) et examen semestriel (moitié de la note).

Contenu (mots-clefs)

1. **Logique combinatoire** : représentations numériques et opérations, fonctions logiques combinatoires, multiplexeurs et démultiplexeurs;
2. **Logique séquentielle** : éléments de mémoire et bascules, compteurs synchrones, machines d'état;
3. **Méthodologie de conception et réalisation** : méthodologie de conception, états logiques, circuits logiques programmables.

Support de cours

Script, slides, exercices, laboratoires et mini-projets.

Outils utilisés

Environnement de développement [Mentor HDL-Designer](#), [Mentor Modelsim](#), [Vivado](#).

Particularité d'organisation

Un cours de complément (2 périodes par semaine) est offert à toutes les personnes souhaitant ou ayant besoin d'approfondir leurs connaissances.

Unité d'enseignement 102.2 – Architecture des ordinateurs (ArchOrd)

Périodes/sem	4	Semestre	Printemps
Enseignant-e	Silvan Zahno	Coefficient	4

Description courte / objectifs

En partant des bases acquises dans le cours 102.1 – *Systèmes numériques*, ce cours aborde les connaissances de base du fonctionnement et de l'organisation d'un processeur, notamment dans ses aspects de fonctionnement logique, de structure ainsi que du point de vue de la performance. Ces nouvelles compétences seront mises à l'épreuve au sein de groupes de travail en laboratoire et dans le cadre d'un projet semestriel.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

3 notes durant le semestre dont le poids est précisé en début de semestre. Cette unité de cours n'a pas d'examen de semestre.

Contenu (mots-clefs)

- **Architecture des processeurs** : structure interne, possibilités d'implémentation, microarchitectures et *Instruction Set Architectures*
- **Calculs de performance** : benchmark, métriques standard
- **Architecture RISC-V** : implémentation et assembleur

Support de cours

Script, slides, exercices, laboratoires et mini-projets.

Outils utilisés

- Langage de description du matériel VHDL
- *SiFive Freedom SDK*, *Ripes*
- Environnement de développement *Mentor HDL-Designer*, *Mentor Modelsim*, *ISE/Vivado*, *Lattice Diamond*

Bibliographie

- J. Hennessy, D. Patterson, *Computer Architecture, Sixth Edition - A Quantitative Approach*, 978-0-12-811905-1, 2019.
- J. Hennessy, D. Patterson, *Computer Organization and Design, RISC-V Edition, 2nd Edition*, 978-0-12-820331-6, 2021.
- Sarah L. Harris, D. M. Harris, *Digital Design and Computer Architecture, RISC-V Edition*, 978-0-12-820064-3, 2022.

103 – Sécurité et communication

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Base des TIC	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Sécurité et communication	Code du module	103
Type	Obligatoire	Crédits ECTS	6
Langue	Français	Année académique	Première année
Responsable(s)	Dr Louis Lettry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
103.1	RIP	Réseaux IP	4						
103.2	CrySec	Cryptographie et sécurité		4					

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	100 h	50 h	150 h

2. Description du module

Le développement d'Internet et de la communication a mis en réseau toutes les machines autour de la planète, ouvrant ainsi la porte à la transformation digitale de la société. Dans ce module, vous verrez comment ces réseaux IP fonctionnent et comment les machines y communiquent. De plus, vous découvrirez également quels sont les risques liés à ces communications et comment s'en prémunir grâce aux techniques de cryptographie et sécurité.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☐ Avoir suivi le(s) module(s)
- ☒ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- identifier les principales problématiques liées au transport sécurisé des données sur un réseau de communication IP **(C)**
- expliquer et détailler les notions de *confidentialité*, *intégrité* et *disponibilité* des données **(C)**

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

- décrire et différencier les différentes techniques de cryptage **(C)**.
- décrire la topologie d'un réseau de communication et en ressortir les principales caractéristiques techniques et liées à sa performance **(A)**
- opposer différents protocoles de communication sécurisés, en décrire les limites et les contraintes **(A)**

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{3 \cdot m_{\text{Rip}} + 3 \cdot m_{\text{CrySec}}}{6}$$

avec :

- m_{Rip} – moyenne des notes en *Réseaux IP*;
- m_{CrySec} – moyenne des notes en *Cryptographie et sécurité*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 103.1 – Réseaux IP (RIp)

Périodes/sem	4	Semestre	Automne
Enseignant·e·s	Dr Gregory Mermoud, Dr Davide Calvaresi	Coefficient	3

Description courte / objectifs

Grâce à ce cours, l'étudiant·e est capable d'expliquer et d'appliquer les principes des réseaux informatiques basés TCP/IP, notamment :

- Comprendre les protocoles de l'Internet et des réseaux informatiques et leur utilisation dans un contexte professionnel.
- Configurer, dépanner et sécuriser des réseaux domestiques ou de petites et moyennes entreprises.
- Identifier les capacités et limitations de l'Internet et des réseaux informatiques du point de vue de l'application.
- Établir des liens entre les principes fondamentaux des réseaux et d'autres disciplines de l'informatique.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

2 notes durant le semestre et 1 examen semestriel. Le poids pour chaque note est indiqué en début de semestre.

Contenu (mots-clefs)

- **Topologies et architectures des réseaux**
- **Modèle ISO** : définition, couches et communication entre elles, protocoles, données, services
- **Couche physique** : débits, canaux, supports physiques et interfaces, multiplexage
- **Adressage** : IPV4, classes, MAC, DNS, introduction à IPV6
- **Liaison et protocoles** : couches liaison de données, routage, réseau et transport
- **Éléments de réseaux** : hubs, routeurs, switch, serveurs d'adresse, VLAN
- **Transport layer** : UDP, TCP, fragmentation, NAT (si le temps le permet)
- **Application layer** : DHCP, DNS, SMTP, Web services, HTTP, proxies
- **Modèle de programmation** : `sockets` et communication client-serveur

Support de cours

Slides, laboratoires et mini-projets.

Outils utilisés

- Outils divers [Wireshark](#), [Cisco Packet tracer](#), [curl](#), [Kali Linux](#).

Bibliographie

- L. Peterson, B. Davie, *Computer Networks: A Systems Approach*, 5th edition, ISBN 978-0123850591

Unité d'enseignement 103.2 – Cryptographie et sécurité (CrySec)

Périodes/sem	4	Semestre	Printemps
Enseignant·e·s	Dr Louis Lettry, Dr Davide Calvaresi, Xavier Barmaz	Coefficient	3

Description courte / objectifs

Les objectifs de ce cours sont d'expliquer et appliquer les principes de la cryptographie et de la sécurité des réseaux par l'acquisition de compétences dans quatre catégories :

1. Algorithmes et théorie de la cryptographie;
2. Application des algorithmes;
3. Techniques et algorithmes de gestion de clés, distribution des clés (SSL) et authentification des utilisateur·trice·s;
4. Usage des algorithmes de chiffrement dans les protocoles réseaux et dans les applicatifs (notamment HTTPS et SSH);
5. Techniques d'attaque et de sécurisation contre les intrus et les logiciels malveillants.

Méthode(s) d'enseignement

- | | | |
|--|---|---|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☒ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre (partie cryptographie), et 1 examen semestriel (partie sécurité). Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Cryptographie** : algorithmes symétriques ou non, clés publiques et distribution de clés, authentification, signatures digitales
- **Sécurité** : confidentialité / intégrité / authentification, mécanismes d'authentification, protocoles réseaux sécurisés, sécurité applicative
- **Laboratoires** : exercices, cryptographie, authentification, sécurité des réseaux et analyse, sécurité Web

Support de cours

Slides, vidéos, exercices, laboratoires.

Outils utilisés

Langage Python, outils de virtualisation.

Bibliographie

- William Stallings, *Cryptography and Network Security, 7th edition*, ISBN-13 978-0134444284, 2016.
- Behrouz A. Forouzan, *TCP/IP Protocol Suite (chapter Cryptography and Network Security)*, ISBN 0073376043, 2009.

Particularité d'organisation

Ce cours est réparti entre les deux enseignant·e·s (une partie cryptographie et une partie sécurité).

104 – Humanités 1

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Humanités et société	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Humanités 1	Code du module	104
Type	Obligatoire	Crédits ECTS	9
Langue	Français, Anglais	Année académique	Première année
Responsable(s)	Pierre-André Mudry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
104.1	Ang1	Anglais 1	4						
104.2	Ang2	Anglais 2		2					
104.4	ComEx	Communication et expression		3					

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	110 h	120 h	230 h

2. Description du module

Ouvertes sur le monde, curieuses, agiles, créatives... dans un monde en constante mutation, les personnes ayant choisi une carrière en informatique doivent – en plus de leurs compétences scientifiques et technologiques – posséder les *soft-skills* leur permettant d'agir avec aisance et pertinence dans la société. Le module *Humanités 1* vise ainsi à développer des compétences transversales en communication orale et écrite (en français et anglais).

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☐ Avoir suivi le(s) module(s)
- ☒ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- communiquer à l'écrit et à l'oral en français et anglais dans des situations du monde du travail **(C)**
- constater l'importance de la langue anglaise dans les milieux professionnels et scientifiques **(C)**
- développer les méthodes permettant de travailler en équipe **(A)**
- développer ses capacités d'analyse, de synthèse et de structuration dans la communication **(A)**

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{4 \cdot m_{\text{Ang1}} + 2 \cdot m_{\text{Ang2}} + 3 \cdot m_{\text{ComEx}}}{9}$$

avec :

- m_{Ang1} – moyenne des notes en *Anglais 1*;
- m_{Ang2} – moyenne des notes en *Anglais 2*;
- m_{ComEx} – moyenne des notes en *Communication et expression*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 104.1 – Anglais 1 (Ang1)

Périodes/sem	4	Semestre	Automne
Enseignant·e·s	Erika Fauchère, Dana Mumford Jacquier	Coefficient	4

Description courte / objectifs

Ce cours vise à améliorer le niveau d'anglais de manière à pouvoir être capable de comprendre et de s'exprimer en anglais dans un environnement académique et professionnel.

Il vise également à pratiquer et à réviser certains points importants de la grammaire et du vocabulaire afin que les étudiant·e·s puissent se sentir plus à l'aise pour communiquer efficacement en anglais. À la fin de ce cours, les étudiant·e·s seront capables de :

- participer à un entretien d'embauche
- faire des présentations plus académiques/professionnelles en anglais
- présenter/défendre un argument participer à des discussions

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Contrôle continu. Le contrôle continu comprend des exposés, des épreuves orales ou écrites.

Contenu (mots-clefs)

- Saisir l'essentiel lorsque la langue standard est utilisée sur des sujets familiers concernant le travail, les sujets d'actualité, l'école, le temps libre, etc.
- Interagir dans différentes situations rencontrées dans le cadre professionnel et académique (TP, séance, projets, etc.)
- Présenter un sujet / projet en utilisant un vocabulaire adapté et un support visuel clair et structuré (oral et écrit)

Support de cours

Matériel fourni par l'enseignant·e.

Outils utilisés

Quizlet, Padlet, Mentimeter

Bibliographie

- M. Foley, D. Hall, *MyGrammarLab advanced C1/C2 without key*, Pearson, ISBN-13 978-1-4082-9912-8, 2012.

Unité d'enseignement 104.2 – Anglais 2 (Ang2)

Périodes/sem	2	Semestre	Printemps
Enseignant·e	Erika Fauchère	Coefficient	2

Description courte / objectifs

Ce cours vise à développer des compétences en communication professionnelle et académique en anglais. À la fin de ce cours, les étudiant·e·s seront capables de :

- présenter des projets de leur domaine d'étude de manière professionnelle
- communiquer efficacement à un niveau managérial et professionnel
- participer avec confiance à des réunions sur le lieu de travail ou à des travaux dirigés et des séminaires
- s'exprimer avec plus d'aisance

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Contrôle continu (au moins 3 notes). Le contrôle continu comprend des exposés, des épreuves orales ou écrites. Chacune des épreuves a le même poids.

Contenu (mots-clefs)

Différentes activités permettant aux étudiant·e·s d'améliorer et de pratiquer leurs compétences en matière de compréhension orale et écrite, production orale et écrite, ainsi que de grammaire et de vocabulaire, à un niveau C1.

Support de cours

Matériel fourni par l'enseignant·e

Outils utilisés

Quizlet, Padlet, Mentimeter

Bibliographie

- M. Foley, D. Hall, *MyGrammarLab advanced C1/C2 without key*, Pearson, ISBN-13 978-1-4082-9912-8, 2012.

Unité d'enseignement 104.4 – Communication et expression (ComEx)

Périodes/sem 3**Enseignant-e** Laurence Laffargue-Rieder**Semestre** Printemps**Coefficient** 3

Description courte / objectifs

Cette unité de cours est destinée à sensibiliser les étudiant-e-s à l'importance de la communication orale et écrite dans la vie professionnelle. L'étudiant-e va dans ce cours améliorer son aptitude à communiquer et convaincre, mettre en pratique les règles de communication (verbales ou non) et être capable de rédiger différents types de plans en fonction de l'information à transmettre.

Les différents éléments théoriques sont appliqués au travers d'ateliers pratiques donnés par des spécialistes du domaine.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Selon les instructions de l'enseignant-e.

Contenu (mots-clefs)

1. Communication et relations interpersonnelles

- rôles et enjeux (public cible, définir les objectifs de la communication)
- au sein de l'équipe, enjeux de groupe (répartition du travail, gouvernance horizontale)
- gérer les conflits
- l'information: bibliographie, qualité des sources, citer ses sources (APA), le plagiat et *ChatGPT*
- développer des stratégies pour communiquer : exprimer son opinion, vulgariser

2. Communication orale

- échanger des idées et développer sa pensée critique
- synthétiser des informations
- convaincre
- networking et small talk
- négocier et pitcher
- auto-analyse et pratique réflexive (se filmer, se voir)

3. Ateliers

- Divers ateliers seront réalisés autour de la communication orale et écrite avec des spécialistes (théâtralisation, marketing, vidéo etc.).

Support de cours

Définis par l'enseignant-e.

Particularité d'organisation

Le cours fera également appel à des intervenant-e-s externes pour animer certains ateliers.

105 – Summer school 1

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Intégration	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Summer school 1	Code du module	105
Type	Obligatoire	Crédits ECTS	4
Langue	Français, Anglais	Année académique	Première année
Responsable(s)	Dr Louis Lettry, Erika Fauchère	Organisation	Par bloc

Volume de travail	Enseignement	Travail personnel	Total
	30 h	90 h	120 h

2. Description du module

Cette première *summer school* est un projet d'intégration autour des données qui vous permettra de mettre en action toutes les connaissances acquises durant la 1^{ère} année par un projet de groupe conséquent, réalisé durant trois semaines. Vous aurez également l'occasion de mettre en pratique les techniques de communication durant différents ateliers de pitches et de présentations, en anglais.

Concrètement, le projet à réaliser est constitué de plusieurs *data challenge* réalisés en petits groupes dans lequel vous allez découvrir une introduction à la science des données. Vous plongerez dans différents ensembles de données concrets, vous devrez choisir des moyens de les décrire et de les manipuler grâce à des outils de calcul scientifique de haut niveau comme [Numpy](#), [pandas](#), ou encore [matplotlib](#) par exemple. En termes de communication, vous travaillerez notamment sur les techniques de présentation et communication orale (choisir et mettre de l'ordre dans ses idées). Vous aurez également l'occasion d'échanger des idées et développer votre pensée critique (en anglais), le tout dans une mise en pratique sur le projet.

Remarque : la présence aux activités de la *summer school* est obligatoire pour pouvoir valider le module.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 101
- ☐ Pas de prérequis
- ☒ Autre : si certains modules n'ont pas été acquis durant l'année, la participation se fait après discussion avec le responsable de filière.

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- utiliser des librairies pour analyser des données disponibles et concevoir des modèles (**C**)

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

- gérer et développer le projet (**A**)
- communiquer dans un contexte professionnel (**A**)
- mettre en application divers savoirs de manière pratique (**A**)
- développer la pensée *ingénieur-e* par la réflexion individuelle et en groupe (**A**)

5. Modalités d'évaluation et de validation

Il n'y a pas de note(s) dans ce module car il est évalué à l'aide d'une échelle *Acquis* | *Non acquis*.

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

Si le module n'est pas acquis, l'étudiant-e doit le refaire ou effectuer un complément (dans les cas limites discutés avec le/la RF).

200 – Sciences IT 2

Descriptif de module

Filière Informatique et systèmes de communication
Axe d'enseignement Sciences de l'ingénierie
Année académique 2025-2026

Formation Bachelor
Orientation Tronc commun

1. Organisation du module

Module	Sciences IT 2	Code du module	200
Type	Obligatoire	Crédits ECTS	6
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr Jessen Page	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
200.1	ComPhy1	Computational physics 1				4			
200.2	ProbStatIT	Probabilités et statistiques IT				4			

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	100 h	70 h	170 h

2. Description du module

La physique et les mathématiques permettent de créer des modèles de certains phénomènes. L'implémentation de ces modèles dans le contexte informatique requiert une compréhension de ces modèles et des méthodes computationnelles sous-jacentes.

L'objectif de ce module est ainsi **d'identifier, analyser et proposer des solutions** à ces problématiques par l'étude de ces méthodes. Des cas d'étude, respectivement sous la forme "laboratoires virtuels" (dans le cas de *Computational physics 1*) et de mini-projets (pour *Probabilités et statistiques*) permettront d'illustrer les différentes thématiques traitées et les mettre en pratique.

3. Prérequis

- ☒ Avoir validé le(s) module(s): 100
- ☒ Avoir suivi le(s) module(s): 101
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

- adopter certaines méthodes utilisées en physique pour modéliser le réel (par ex. systèmes mécaniques, thermiques, électriques, fluidiques ainsi que biologiques, sociaux et économiques) (**A**)
- choisir un type de modèle et le paramétrer grâce à l'analyse d'un jeu de données du système à modéliser (**J**)
- appliquer les méthodes mathématiques analytiques et numériques vues en cours (**A**)
- évaluer la qualité d'un jeu de données pour des utilisations futures grâce à des tests statistiques (**A**)
- afficher de façon judicieuse des indicateurs de l'analyse d'un jeu de données (**C**)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{3 \cdot m_{\text{ComPhy1}} + 3 \cdot m_{\text{ProbStatIT}}}{6}$$

avec :

- m_{ComPhy1} – moyenne des notes en *Computational physics 1*;
- $m_{\text{ProbStatIT}}$ – moyenne des notes en *Probabilités et statistiques IT*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant·e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant·e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint·e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 200.1 – Computational physics 1 (ComPhy1)

Périodes/sem	4	Semestre	Automne
Enseignant·e·s	Dr Jessen Page, Dr Florian Desmons	Coefficient	3

Description courte / objectifs

L'ingénieur·e en informatique a souvent pour mandat de transformer un jeu de données en un autre ou de participer à la réalisation de cette transformation. Derrière chaque transformation se cache un modèle.

L'objectif premier de la physique est d'expliquer le comportement du réel grâce à un ensemble de modèles, paramétrés et validés par des données mesurées. Très souvent, ces modèles reposent sur des méthodes de calcul numérique et des implémentations informatiques (modèles de simulation).

Ce cours a pour objectif de familiariser les étudiant·e·s aux approches utilisées en physique pour modéliser "le réel" ainsi qu'aux méthodes mathématiques analytiques et numériques sur lesquelles elles reposent.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Mécanique newtonienne et pré-newtonienne** : dynamique des astres, force de gravitation, modèles paramétriques empiriques
- **Oscillations harmoniques** : résolution de systèmes d'équations différentielles, intégration numérique, oscillateurs harmoniques couplés, systèmes chaotiques
- **Flux** : transfert de chaleur stationnaire et transitoire, diffusion, méthodes numériques à différences finies, lois de Kirchhoff et résolution de systèmes d'équations linéaires

Support de cours

- Documents mis à disposition par les professeur·e·s sous format numérique
- Séries, laboratoires et ressources en ligne

Outils utilisés

- Langage de programmation [Julia](#)
- Notebooks [Pluto](#)

Particularité d'organisation

Chaque thématique (durant quatre semaines environ) est suivie par un laboratoire sur ordinateur.

Unité d'enseignement 200.2 – Probabilités et statistiques IT (ProbStatIT)

Périodes/sem	4	Semestre	Automne
Enseignant·e·s	Dr Cédric Travelletti, Dr Florian Desmons	Coefficient	3

Description courte / objectifs

La statistique fournit des méthodes permettant de représenter et d'interpréter des données, dans le but de les valoriser et d'en extraire les caractéristiques essentielles. Ces méthodes sont basées sur des modèles mathématiques issus des probabilités. L'objectif de ce cours est de familiariser les étudiant·e·s avec les modèles et les outils statistiques de base afin de pouvoir les appliquer pour analyser des données.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Deux notes au cours du semestre (un projet et un examen écrit) et un examen oral semestriel. La pondération de chaque évaluation est précisée au début du semestre.

Contenu (mots-clefs)

- **Bases de probabilités** : ensembles, indépendance, probabilité conditionnelle, théorème de Bayes
- **Statistique descriptive** : indicateurs statistiques, histogramme, box-plot
- **Variables aléatoires** : lois de probabilité discrètes et continues
- **Statistique inférentielle** : intervalles de confiance, tests statistiques
- **Modèles de régression** : régression linéaire simple et multiple (si le temps le permet), analyse de variance (ANOVA)
- **Réduction de dimensionnalité** : analyse en composantes principales

Support de cours

- Documents mis à disposition par les professeur·e·s sous format numérique
- Séries d'exercices et laboratoires

Outils utilisés

- Langage de programmation [Julia](#)
- Notebooks [Pluto](#)

Particularité d'organisation

Ce cours est réparti entre les deux enseignants, un pour la partie théorique et exercices et l'autre pour la partie numérique et mini-projet.

201 – Fondements de la programmation 2

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Base des TIC	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Fondements de la programmation 2	Code du module	201
Type	Obligatoire	Crédits ECTS	12
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr Louis Lettry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
201.1	Algo	Algorithmes et structures de données			4				
201.2	SoftEng	Génie logiciel			4				
201.3	DBRel	Bases de données relationnelles			4				

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	150 h	150 h	300 h

2. Description du module

Le design de tout logiciel en ingénierie informatique procède de quatre étapes essentielles.

La première consiste à *déterminer les besoins en termes de fonctionnalités d'un logiciel*. A partir de ces besoins, il est alors possible de procéder à la conception de l'architecture du logiciel ainsi que de ses composants, ce que se propose de faire l'unité d'enseignement 201.1 *Génie logiciel*.

La deuxième étape cruciale consiste à *comprendre, choisir, designer et implémenter* des fonctionnalités du logiciel à l'aide des algorithmes et des structures de données adéquates. Ce sont ces thématiques qui seront au cœur de l'unité *Algorithmes et structures de données*.

La troisième consiste à structurer l'information de manière pratique et élégante afin qu'elle puisse être utilisée de manière adéquate, ce qui sera abordé dans l'unité *Bases de données relationnelles* en faisant un lien entre les données brutes et les données structurées.

La dernière étape, à l'interface entre ces unités, réside dans le *test des fonctionnalités et des performances* qui sera réalisé selon les outils spécifiques à chacune des branches.

3. Prérequis

- ☒ Avoir validé le(s) module(s): 101
- ☒ Avoir suivi le(s) module(s): 103
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

- spécifier des besoins logiciels et implémenter ces spécifications selon les standards du métier (**A**)
- évaluer des designs de logiciels en termes de structures et algorithmes (**C**)
- connaître les structures de données et les algorithmes standard associés (**C**)
- choisir et appliquer les méthodes algorithmiques de minimisation de l'espace de recherche (**A**)
- concevoir et implémenter des algorithmes pour résoudre de nouveaux problèmes (**C**)
- analyser, quantifier et optimiser les performances d'un algorithme ou programme donné (**C**)
- énumérer les aspects techniques des BD et les défis d'une mise en production (**C**)
- modéliser des systèmes simples à l'aide du langage SQL (**A**)
- relier les concepts et techniques apprises dans ce module avec les autres concepts clés des cours ISC (**C**)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{4 \cdot m_{\text{Algo}} + 4 \cdot m_{\text{SoftEng}} + 4 \cdot m_{\text{DBRel}}}{12}$$

avec :

- m_{Algo} – moyenne des notes en *Algorithmes et structures de données*;
- m_{SoftEng} – moyenne des notes en *Génie logiciel*;
- m_{DBRel} – moyenne des notes en *Bases de données relationnelles*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
☒ Pas de remédiation
☐ Autre

7. Modalités de répétition

L'étudiant·e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant·e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint·e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

Unité d'enseignement 201.1 – Algorithmes et structures de données (Algo)

Périodes/sem	4	Semestre	Automne
Enseignant-e-s	Dr Émilie Neveu, Dr Louis Lettry	Coefficient	4

Description courte / objectifs

Ce cours fournit une introduction à la **modélisation** mathématique de problèmes computationnels. Il couvre les algorithmes les plus courants, les différentes catégories d'algorithmes ainsi que les structures de données nécessaires à résoudre ces problèmes. S'appuyant notamment sur le contenu de *100.5 - Maths discrètes*, ce cours met l'accent sur la relation algorithme / programmation et introduit des techniques d'analyse de performance de ces problèmes.

Le cours présente une introduction sur la théorie des automates et de l'algorithmique (complexité et optimisation). Cette introduction sera suivie par un approfondissement des bons usages de différentes structures de données déjà rencontrées précédemment.

Cette première partie permettra d'ouvrir aux techniques de recherche et d'exploration d'espace afin de trouver une solution à un problème d'optimisation ou de recherche de zéro d'une fonction continue. Différentes méthodes d'optimisation algorithmique classiques seront abordées (*backtracking* et programmation dynamique) afin de minimiser le nombre d'états de recherche à visiter et réduire temps et mémoire. Finalement, le cours sera ponctué par un projet, évalué en groupe, qui sera accompagné d'un petit tournoi entre les participant-e-s.

Méthode(s) d'enseignement

- | | | |
|--|---|---|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☒ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Algorithmes et calculabilité** : théorie des automates, $\mathcal{P}$ vs $\mathcal{NP}$, Turing completeness;
- **Complexité algorithmique**: temps, mémoire et notation $\mathcal{O}$
- **Structures de données linéaires** : tableaux, listes, queues, piles
- **Théorie des graphes** : profondeur, largeur, Dijkstra
- **Optimisation algorithmique** : backtracking, branch and bound, programmation dynamique
- **Approximations** : heuristiques stochastiques, A^* si le temps le permet

Support de cours

Slides, exercices en ligne, laboratoires et mini-projets.

Outils utilisés

Langage [Python](#).

Bibliographie

- Cormen, T. H. *et al.*, *Introduction to algorithms*, 3rd edition, ISBN-13 978-0262033848, 2009.
- Meyer B., *Touch of Class*, ISBN-13 978-3540921448, 2013.
- Wack B. *et al.*, *Informatique pour tous en classes préparatoires aux grandes écoles*, Eyrolles, 2013.

Particularité d'organisation

Le planning du cours sera communiqué en début de semestre, avec notamment la répartition détaillée des cours selon les professeur·e·s.

Unité d'enseignement 201.2 – Génie logiciel (SoftEng)

Périodes/sem	4	Semestre	Automne
Enseignant·e·s	Dr Michael Schumacher, Dr Davide Calvaresi	Coefficient	4

Description courte / objectifs

Ce cours enseigne les bases du développement logiciel à l'aide des outils de l'ingénierie logicielle : concevoir des logiciels, écrire du bon code, et livrer du logiciel. Il met en avant les techniques permettant d'obtenir un code de qualité, robuste, modulable, gérable et évolutif. À la fin du cours, l'étudiant·e est capable de :

- écrire du code qui fonctionne selon les spécifications
- écrire du code qui soit maintenable
- s'organiser en équipe pour réaliser un projet de taille moyenne
- évaluer des choix de design de logiciel et proposer des alternatives

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Contrôle continu avec 2 notes durant le semestre. Les deux notes ont le même poids.

Contenu (mots-clefs)

1. Cycles de vie du développement logiciel

- Introduction au cycle de vie du développement logiciel et aux méthodes de gestion de développement
- Techniques de spécification des besoins (*requirements engineering*)

2. Modélisation avec UML (*Unified Modelling Language*)

3. Développement d'architectures logicielles modulables et évolutives

- Principes de design
- Design patterns classiques et combinaisons de patterns
- Bad Smells et refactoring

4. Gestion de la qualité du logiciel

- Tests
- Analyse de qualité, débogage
- Performance (analyse de code)

5. User eXperience et User Interfaces

- Concepts de base, requirement, perception
- prototypage, design d'alternatives et évaluation

6. Gestion de projets de développement agiles

- Introduction aux processus de développement agiles
- Processus de développement agile SCRUM : valeurs, rôles, rituels, artefacts

Support de cours

Slides, lectures, vidéos, exercices, laboratoires et mini-projets.

Outils utilisés

- Langage de programmation pour les designs patterns : Java, [version 17](#)
- Environnement de développement [IntelliJ](#)
- Design UML avec [IntelliJ Diagrams](#), et autre outil à définir

Bibliographie

- E. Freeman, E. Robson, *Design Patterns: Head First Design Patterns, 2nd edition*, ISBN-13 978-1492078005, 2021.
- E. Gamma *et al.*, *Design Patterns: Elements of Reusable Object-Oriented Software*, ISBN-13 978-0201633610, 1994.
[disponible en ligne](#)
- *The refactoring website* <https://refactoring.guru/refactoring>.
- Martina Seidl *et al.*, *UML@Classroom*, Springer Verlag, ISBN-13 978-3319127415, 2015.

Particularité d'organisation

Le planning du cours sera communiqué en début de semestre, avec notamment la répartition détaillée des cours selon les professeur·e·s.

Unité d'enseignement 201.3 – Bases de données relationnelles (DBRe1)

Périodes/sem	4	Semestre	Automne
Enseignant-e	Dr Renaud Richardet	Coefficient	4

Description courte / objectifs

Les bases de données (DB) permettent d'organiser et stocker de l'information de manière à pouvoir l'accéder facilement par des programmes. Ainsi, la plupart des applications informatiques utilisent une base de données pour stocker de manière durable, fiable et efficaces leurs données.

Il existe plusieurs organisations possibles pour les DB et l'une des plus répandues consiste à organiser des données dans des tables qui sont reliées entre elles : c'est le modèle relationnel. Derrière cette apparente simplicité se cache en réalité tout un monde de possibles qui seront traités dans ce cours.

L'objectif principal de ce cours est ainsi de permettre d'appréhender les principaux concepts des bases de données relationnelles en suivant une approche pratique. Les sujets principaux sont la modélisation selon le modèle relationnel, le langage SQL et les opérations sur les DB. Le cours aborde également des aspects plus techniques des DB, tels que l'optimisation et les défis d'une mise en production.

Objectifs :

A la fin du cours, l'étudiant-e sera capable de :

- comprendre le rôle des DB dans la pratique
- maîtriser les bases du modèle relationnel et connaître les niveaux de normalisation et les opérateurs de base
- appliquer des opérations sur les données, à l'aide du langage SQL : création de DB et de tables, définition de schémas avec les types appropriés, ingestion de données, transactions, requêtes simples et plus complexes comprenant des joints et des comptages
- modéliser des systèmes simples à l'aide du langage SQL
- énumérer les aspects techniques des DB et les défis d'une mise en production
- relier les concepts et techniques apprises durant ce cours avec les autres cours ISC passés, présents et futurs

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Contrôle continu avec 2 notes durant le semestre. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Modèle relationnel** : tables, clés primaire et étrangère, index, normalisation
- **Langage SQL** : création de bases et tables, schémas, requêtes simples et complexes, transactions
- **Modélisation de systèmes**
- **Aspects techniques** : optimisation, mise en production

Support de cours

Documents mis à disposition par les professeur·e·s sous format numérique. Séries d'exercices ou laboratoires et ressources en ligne.

Outils utilisés

- PostgreSQL
- Beekeeper Studio, community edition

Bibliographie

- C. Fehly, *SQL Visual quick guide*, 3rd edition, ISBN-13 978-0321553577, 2008.

202 – Systèmes et concurrence

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Base des TIC	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Systèmes et concurrence	Code du module	202
Type	Obligatoire	Crédits ECTS	6
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr Dimi Racordon	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
202.1 OS	Systèmes d'exploitation				4				
202.2 ProCon	Programmation concurrente				2				

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	75 h	75 h	150 h

2. Description du module

A la suite du modèle d'exécution permettant la réalisation de programmes impératifs à un seul processus de la 1ère année, ce module étend les connaissances en rajoutant les processus concurrents. Au travers des principes des systèmes d'exploitation, vous découvrirez ainsi comment un processeur peut gérer la concurrence en termes d'exécution (*threads*, *processes*) mais également de mémoire. Vous découvrirez les différents mécanismes de bases permettant de gérer cette concurrence du point de vue système (en langage C++) mais également grâce aux différents outils de la programmation concurrente (en Java).

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 101, 102, 103
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- utiliser les concepts clés des systèmes d'exploitation (mémoire, tâches, *threads*, virtualisation ...) (C)

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (C) Connaissances et compréhension, (A) Application, (J) Jugement (analyse, synthèse, évaluation).

- appliquer les méthodes efficaces de structuration des programmes concurrents (C)
- connaître le modèle relationnel et l'appliquer de manière concrète (C)
- relier les concepts et techniques apprises dans ce module avec les autres concepts clés des cours ISC (C)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{4 \cdot m_{OS} + 2 \cdot m_{ProCon}}{6}$$

avec :

- m_{OS} – moyenne des notes en *Systèmes d'exploitation*;
- m_{ProCon} – moyenne des notes en *Programmation concurrente*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant·e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant·e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint·e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 202.1 – Systèmes d'exploitation (OS)

Périodes/sem	4	Semestre	Automne
Enseignant-e	Dr Dimi Racordon	Coefficient	4

Description courte / objectifs

Les systèmes d'exploitation offrent une couche d'abstraction uniforme au-dessus du matériel pour en simplifier l'usage. En particulier, ils proposent une interface permettant de contrôler l'allocation des processeurs et de la mémoire, et d'orchestrer les interactions avec les périphériques. Ils forment donc ainsi une partie essentielle de tout système informatique.

L'objectif de ce cours est de présenter les concepts fondamentaux des systèmes d'exploitation au travers d'une approche pratique. Les sujets abordés comprennent notamment la gestion de processus, l'organisation de la mémoire vive et mémoire de stockage, et la programmation système. A la fin du cours, l'étudiant-e sera capable de décrire les principes clefs des systèmes d'exploitation modernes et d'utiliser leurs interfaces pour interagir avec les ressources d'un système informatique.

En outre, l'étudiant-e sera sensibilisé-e aux défis que relèvent les accès directs à la mémoire dans les applications de hautes performance et/ou concurrentes.

Objectifs :

- être capable de décrire les concepts clefs des systèmes d'exploitation: pagination mémoire, fichiers, processus d'exécution, ordonnancement;
- être capable d'utiliser un shell pour interagir avec un gestionnaire de processus;
- être capable d'utiliser l'interface d'un système d'exploitation pour démarrer des processus et allouer de la mémoire en C++;
- être capable d'expliquer les problèmes majeurs liés à la programmation concurrente: interblocage, compétition, famine.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Une note issue d'un examen écrit en cours de semestre complétée par les évaluations des laboratoires.. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Programmation système** : `bash` et C++
- **Gestion de la mémoire** : pagination, segmentation, swapping
- **Persistance mémoire** : système de fichiers
- **Allocation de processus** : planification, ordonnancement
- **Communication entre processus** : pipe
- **Concurrence** : exclusion mutuelle, synchronisation

Support de cours

- Documents mis à disposition par les professeur-e-s sous format numérique
- Séries d'exercices, laboratoires et ressources en ligne

Outils utilisés

- Environnement de développement [VSCode](#)
- Environnement Linux, macOS ou WSL2

Bibliographie

- Anderson T. and Dhalin M., *Operating Systems: Principles and Practice*, 2nd edition, ISBN-13 978-0985673529, 2022.
- Arpaci-Dusseau et al., *Operating Systems: Three Easy Pieces*, ISBN-13 978-1985086593, 2018.
- Clay Breshears, *The Art of Concurrency: A Thread Monkey's Guide to Writing Parallel Applications*, 1st Edition, ISBN-13 978-0596521530, 2009.

Particularité d'organisation

Le cours débute au début du semestre et dure 4/6 du semestre.

Unité d'enseignement 202.2 – Programmation concurrente (ProCon)

Périodes/sem	2	Semestre	Automne
Enseignant-e	Dr Michael Schumacher	Coefficient	2

Description courte / objectifs

Ce cours vise à présenter les principaux concepts de la programmation concurrente (*thread*, exclusion mutuelle, communication et synchronisation) et les outils et bibliothèques de haut niveau (frameworks d'exécution de *threads*, divers objets de verrouillage, synchroniseurs, ...). Grâce aux notions abordées, l'étudiant-e sera capable de comprendre et fournir des solutions aux principaux défis qui se posent lors de la conception et de la mise en œuvre de systèmes concurrents.

Objectifs :

- situer le domaine de programmation concurrente parmi les autres techniques de programmation et comprendre les problématiques liées à l'exécution concurrente de processus
- manipuler les principales primitives de gestion de la concurrence
- comprendre les méthodes efficaces de structuration des programmes concurrents

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins une note. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- Objets *threads* et synchronisation
- Liveness, guarded blocks et objets immuables, objets *lock*
- Exécuteurs, *callables*, *fork/join*
- Collections concurrentes, variables atomiques, synchroniseurs
- Débogage de programmes concurrents, chapitre choisis avancés.

Support de cours

- Documents mis à disposition par les professeur-e-s sous format numérique
- Séries d'exercices, laboratoires et ressources en ligne

Outils utilisés

- Environnement de développement [IntelliJ](#)
- Langage de programmation *Java*

Bibliographie

- [Oracle The Java Tutorial - Lesson: Concurrency](#).
- Schild, H., *Java: The Complete Reference*, Twelfth Edition, ISBN 9781260463422, 2021, [online resource](#).

Particularité d'organisation

Le cours dure 2/6 du semestre et débute après le cours *Systèmes d'exploitation*.

203 – Humanités 2

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Humanités et société	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Humanités 2	Code du module	203
Type	Obligatoire	Crédits ECTS	4
Langue	Français, Anglais	Année académique	Deuxième année
Responsable(s)	Dr Pierre-André Mudry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
203.1	EthIT Éthique et droit IT				2				
203.3	EngIT Engineering in computer science				3				

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	64 h	44 h	108 h

2. Description du module

Ouvertes sur le monde, curieuses, agiles, créatives... dans un monde en constante mutation, les personnes ayant choisi une carrière en informatique doivent – en plus de leurs compétences scientifiques et technologiques – posséder les *soft-skills* leur permettant d'agir avec aisance et pertinence dans la société. Le module *Humanités 2* vise à développer des compétences transversales en communication ainsi que sur les questions d'éthique et de droit en lien avec le domaine informatique. **Pour l'année 2024–2025, le module comportera l'anglais de manière transitoire. A partir de 2025–2026, l'anglais sera remplacé par la gestion de projet IT (l'anglais sera fait entièrement en 1ère année).**

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 104
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

- communiquer en anglais à l'oral ainsi qu'à l'écrit dans un contexte professionnel **(A)**

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

- comprendre quelles dispositions légales spécifiques s'appliquent aux technologies **(A)**
- appliquer des raisonnements juridiques simples dans différents domaines du droit liés à l'informatique **(A)**
- connaître les enjeux éthiques liés aux technologies informatiques **(C)**
- connaître les étapes d'un projet informatique et appliquer différentes méthodes agiles pour gérer son déroulement **(A)**

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{2 \cdot m_{\text{EthIT}} + 2 \cdot m_{\text{EngIT}}}{4}$$

avec :

- m_{EthIT} – moyenne des notes en *Éthique et droit IT*;
- m_{EngIT} – moyenne des notes en *Engineering in computer science*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 203.1 – Éthique et droit IT (EthIT)

Périodes/sem	2	Semestre	Automne
Enseignant·e	François Charlet	Coefficient	2

Description courte / objectifs

Ce cours vise à introduire les différentes problématiques juridiques et éthiques liées à l'ingénierie informatique et aux données.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel (oral). Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- 1. Le droit général et le cadre juridique** : principes généraux et sources du droit
- 2. Droit pénal** : concepts de droit pénal général, infractions de droit pénal informatique
- 3. Droit civil** : droits de la personnalité, droit des contrats orienté pour l'IT
- 4. Propriété intellectuelle I** : généralités sur la propriété intellectuelle, droit d'auteur, licences logicielles et *open-source*
- 5. Propriété intellectuelle II** : droit des marques, noms de domaine, brevets logiciels
- 6. Protection des données** : cadres suisse (LPD) et européen (RGPD), mesures organisationnelles et techniques
- 7. Intelligence artificielle et algorithmes** : aspects juridiques et éthiques

Support de cours

Articles, jurisprudence, études de cas.

Bibliographie

Fournie en classe.

Particularité d'organisation

Le cours est donné chaque 2 semaines, avec des études de cas pratiques.

Unité d'enseignement 203.3 – Engineering in computer science (EngIT)

Périodes/sem	3	Semestre	Automne
Enseignant-e	Dr Grégory Mermoud	Coefficient	2

Description courte / objectifs

Ce cours offre une perspective essentielle sur ce que signifie être un ingénieur-e en informatique. Contrairement aux disciplines d'ingénierie traditionnelles — où les contraintes physiques et des principes établis guident la conception de structures tangibles — l'ingénieur-e en informatique opère dans un domaine plus fluide et adaptatif. Nous commencerons par explorer ces différences et similitudes fondamentales, en mettant en avant comment l'ingénierie en informatique doit combiner des méthodes formelles et structurées avec le besoin de flexibilité et d'itération continue.

Sur cette base, nous examinerons ensuite l'évolution des méthodologies d'ingénierie logicielle, en opposant l'approche rigide et séquentielle du modèle Waterfall à la nature itérative et adaptative des méthodes agiles. Nous aborderons les aspects clés de la conception logicielle, notamment la définition d'un cahier des charges, la gestion des changements, la conception et l'architecture logicielle, ainsi que le rôle essentiel de la sécurité, du test logiciel et de l'assurance qualité. Nous discuterons également les promesses et défis de l'utilisation de l'Intelligence Artificielle (IA) dans la conception de logiciels. Nous aborderons également l'agilité sous un angle pratique, en présentant quelques-uns des nombreux outils développés dans les dernières décennies tels que les pipelines **CI/CD**, le contrôle de versioning (**Git**), les outils de gestion de projet (**GitLab**), la gestion des dépendances, le scripting shell, la manipulation de données, les éditeurs, les débogueurs et profileurs, ainsi que les conteneurs.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Selon les instructions de l'enseignant-e.

Contenu (mots-clés)

- Appliquer les concepts d'agilité dans la gestion de projet en utilisant des outils et des méthodes adaptés pour optimiser la collaboration et la productivité.
- Comparer les principes fondamentaux de l'ingénierie informatique avec ceux des disciplines d'ingénierie traditionnelles afin de mieux comprendre les spécificités du domaine.
- Mettre en œuvre les étapes clés du développement logiciel, y compris la définition du cahier des charges, la gestion des changements, l'architecture logicielle et l'assurance qualité.
- Utiliser des outils modernes d'ingénierie logicielle (**Git**, **Github**, **CI/CD**, conteneurs, debuggers, scripting) pour gérer efficacement le cycle de vie du développement.
- Évaluer les principes et pratiques de la sécurité logicielle afin d'identifier et d'atténuer les risques associés au développement logiciel.
- Maîtriser l'utilisation d'outils en ligne de commande (**CLI**), du scripting shell et d'autres automatisations afin d'améliorer l'agilité individuelle et l'efficacité opérationnelle dans le développement logiciel.
- Analyser les implications de l'intégration de l'Intelligence Artificielle dans la conception et la maintenance des logiciels, en évaluant ses avantages et ses limites.

Particularité d'organisation

Cette unité de cours est transversale : les méthodes apprises seront directement appliquées dans et pour d'autres modules.

204 – Sciences IT 3

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientation	Data engineering
Année académique	2024-2025		

1. Organisation du module

Module	Sciences IT 3	Code du module	204
Type	Obligatoire	Crédits ECTS	7
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr Jessen Page	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
204.1	SigInf	Signal and information				4			
204.2	ComPhy2	Computational physics 2				4			

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	100 h	80 h	180 h

2. Description du module

Un rôle essentiel que joue l'informatique est celui d'assister ses utilisateurs et utilisatrices dans la prise de décision («*computer assisted decision making*»). À cette fin, l'outil informatique doit être en mesure de représenter le monde physique de manière appropriée aux problèmes qu'il est sensé résoudre. Ce module aborde ce sujet sous deux angles :

1. le *signal* en tant que support d'information et son analyse numérique permettant l'interaction avec le monde physique (dans l'unité d'enseignement *Signal and information*)
2. le *modèle* comme outil utilisé pour la prise de décision en temps réel (dans l'unité d'enseignement *Physics-inspired modelling*).

Le premier cours présente les principaux concepts et techniques qui permettront aux étudiants de savoir pré-traiter, analyser et déduire les contenus des signaux à différentes dimensions. Dans le second cours, les étudiant·e·s se familiariseront avec des approches empruntées à la physique, les méthodes mathématiques analytiques et numériques sur lesquelles elles reposent et les applications de ces méthodes dans d'autres disciplines.

3. Prérequis

- ☒ Avoir validé le(s) module(s): 100
- ☒ Avoir suivi le(s) module(s): 200
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- adopter certaines méthodes utilisées en physique pour modéliser le réel (**A**)
- comprendre l'application de méthodes issues de la physique dans d'autres disciplines (**C**)
- appliquer les méthodes mathématiques analytiques et numériques vues en cours (**A**)
- utiliser des méthodes d'optimisation à des fins de design et de gestion de systèmes (**A**)
- décrire et utiliser les principaux outils pour déduire et observer les caractéristiques principales d'un signal quelconque (**A**)
- fournir des solutions et des pistes lorsque le signal est perturbé (**A**)
- adapter les méthodes de stockage et de traitement des signaux en fonction du contexte (**A**)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{4 \cdot m_{\text{SigInf}} + 3 \cdot m_{\text{ComPhy2}}}{7}$$

avec :

- m_{SigInf} – moyenne des notes en *Signal and information*;
- m_{ComPhy2} – moyenne des notes en *Computational physics 2*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant-e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

Unité d'enseignement 204.1 – Signal and information (SigInf)

Périodes/sem	4	Semestre	Printemps
Enseignant·e·s	Dr Benedetta Franceschiello, Dr Robin Delabays	Coefficient	4

Description courte / objectifs

Le traitement du signal est une étape cruciale de l'analyse des données, car il permet d'extrapoler l'information et étudier l'état du système dont on mesure le signal. Tout notre monde, notre réalité, est signal : le son, une image, une matrice. Un objectif de ce cours est de présenter les concepts clés du traitement du signal à travers les instruments de la théorie de l'information.

Le cours présente les principaux concepts et techniques qui permettront aux étudiants de savoir pré-traiter, analyser et déduire les contenus des signaux à différentes dimensions.

A la fin du cours, l'étudiant·e sera capable de décrire et utiliser les principaux outils pour déduire et observer les caractéristiques principales d'un signal quelconque, autant que de fournir des solutions et des pistes lorsque le signal est perturbé.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Deux contrôles continus durant le semestre. Le premier comptant pour 50% de la note finale (théorie+code), le second (oral) pour 50%.

Contenu (mots-clés)

- **Théorie de l'information** : bits, digits, Shannon, codage et entropie
- **Compression**
- **Codage optimal** : détection et correction d'erreurs
- **Traitement du signal** : pré-traitement et analyse en espace réel et espace fréquentiel (Fourier) et en n dimensions
- **Bruit** : détection de la présence de bruit et d'artefacts
- **Filtres et statistique du signal**

Support de cours

Documents mis à disposition par les professeur.e.s sous format numérique, séries d'exercices et/ou laboratoires en ligne.

Outils utilisés

Langage `Julia` ou `Python`, selon les préférences.

Bibliographie

- T. Cover, J. Thomas, *Information Theory*, ISBN-13 978-0471241959, 2006.
- O. Gazi, *Information theory for electrical engineers*, ISBN-13 978-9811084317, 2018.
- G. Blanchet, M. Charbit, *Digital signal and image processing using MATLAB*, ISBN-13 978-1848216402, 2006.
- A. A. Bruen et al., *Cryptography, Information Theory, and Error-Correction*, ISBN-13 : 978-1-119-58242-7, 2021.
- La liste des livres utilisés pour la rédaction de ce cours n'est pas exhaustive

Particularité d'organisation

Le cours est séparé en deux intervenant·e·s, chacun·e spécialiste d'un domaine (respectivement traitement du signal à l'aide des instruments issue de la théorie de l'information et mathématiques appliquées).

Unité d'enseignement 204.2 – Computational physics 2 (ComPhy2)

Périodes/sem	4	Semestre	Printemps
Enseignant-e-s	Dr Jessen Page, Dr Florian Desmons	Coefficient	3

Description courte / objectifs

Les modèles issus de la physique ont souvent servi d'inspiration pour le développement de modèles dans une variété de disciplines (recherche opérationnelle, mathématiques financières, théorie de l'information, contrôle...). Ainsi les méthodes mathématiques et numériques sous-jacentes à ces modèles ont percolé d'une discipline à une autre, diversifiant leurs champs d'application.

Dans ce cours, les étudiant-e-s se familiariseront avec quelques approches utilisées en physique pour modéliser le réel, les méthodes mathématiques analytiques et numériques sur lesquelles elles reposent et les applications de ces méthodes dans d'autres disciplines.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Processus stochastiques** : marche aléatoire, méthode de Monte Carlo, mouvement brownien, entropie et réversibilité
- **Mécanique statistique** : théorie cinétique des gaz (variables macroscopiques et microscopiques), gaz parfaits et réels, changement de phase

Support de cours

- Documents mis à disposition par les professeur-e-s sous format numérique
- Séries, laboratoires et ressources en ligne

Outils utilisés

- Langage de programmation [Julia](#)
- Notebooks [Pluto](#)

Particularité d'organisation

Chaque thématique d'importance est suivie d'un laboratoire sur ordinateur.

205 – Advanced programming

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientations	Data engineering
Année académique	2025-2026		

1. Organisation du module

Module	Advanced programming	Code du module	205
Type	Obligatoire	Crédits ECTS	11
Langue	Français, Anglais	Année académique	Deuxième année
Responsable(s)	Dr Pierre-André Mudry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
205.1	FunProg	Functional programming				3			
205.2	BeyDB	Beyond relational databases				4			
205.3	FStack	Full-stack Web development				4			

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	160 h	140 h	300 h

2. Description du module

Il existe tout un univers de technologies au delà de la programmation orientée-objet, de la complexité du partage d'état, de la rigidité des bases de données relationnelles, de la concurrence et des tests bien souvent insuffisants. Ainsi, ce module introduit les technologies les plus avancées du point de vue de la programmation, applicables autant dans le contexte du Web, du *cloud* ainsi que pour le *big data*.

3. Prérequis

- ☒ Avoir validé le(s) module(s): 101
- ☒ Avoir suivi le(s) module(s): 201, 202
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- maîtriser les principes de la programmation fonctionnelle en l'absence d'état (**C**)
- juger les avantages et inconvénients du typage dans le contexte de la programmation et de la structuration des données (**J**)

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

- maîtriser et contraster différents modèles de structuration de données (**J**)
- comprendre les contextes d'utilisations de constructs avancés comme les acteurs et les *futures* (**C**)
- interpréter les enjeux lors de la définition d'API et de la communication inter-programmes, concurrents ou non (**C**)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{3 \cdot m_{\text{FunProg}} + 4 \cdot m_{\text{BeyDB}} + 4 \cdot m_{\text{FStack}}}{11}$$

avec :

- m_{FunProg} – moyenne des notes en *Functional programming*;
- m_{BeyDB} – moyenne des notes en *Beyond relational databases*;
- m_{FStack} – moyenne des notes en *Full-stack Web development*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
☒ Pas de remédiation
☐ Autre

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au ½ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 205.1 – Functional programming (FunProg)

Périodes/sem	3	Semestre	Printemps
Enseignant-e	Dr Pierre-André Mudry	Coefficient	3

Description courte / objectifs

Ce cours vise à introduire le paradigme de programmation fonctionnel, notamment dans le contexte de la concurrence et des *domain-specific languages*.

Vous y verrez notamment comment l'*absence d'état* permet de rapprocher mathématiques et programmation tout en offrant de singuliers avantages pour la concurrence. Le fonctionnel permet également de considérer les fonctions comme des *first class citizen*, offrant la possibilité de s'affranchir en grande partie d'une pensée linéaire du code par le biais des lambdas et des fonctions d'ordre supérieur.

Méthode(s) d'enseignement

- | | | |
|--|---|---|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☒ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre. Le cas échéant, le poids de chaque note est précisé en début de semestre. Cette unité d'enseignement ne comporte pas d'examen semestriel.

Contenu (mots-clefs)

• Introduction

- Concepts et paradigmes des langages de programmation
- Evaluation des expressions.

• Programmation fonctionnelle

- Absence d'état, transparence référentielle, fonctions pures, raisonnement à propos des programmes
- Evaluation avancée ou retardée (*eager vs lazy*)
- Types algébriques, inférence de types et pattern matching
- Fonctions de rang supérieur et récursion
- Une application : un interpréteur pour un petit langage de programmation impératif.

• Programmation multiparadigmes et concurrente

- Réintroduction de l'état en mélangeant programmation fonctionnelle et orientée objet
- Parallélisme sans état
- Gérer la concurrence à l'aide du modèle acteurs.

• Projet applicatif

- Thématiques diverses telles que *domain-specific languages*, agents concurrents, *reactive programming*, ...

Support de cours

Slides, vidéos, exercices en ligne, laboratoires et mini-projets.

Outils utilisés

- Langage de programmation [Scala 2.13 ou 3.X](#)
- Environnement de développement [IntelliJ](#)

Bibliographie

- [Scala reference manual](#).
- Martin Odersky et al., *Programming in Scala, Fifth edition*, ISBN-13 978-0997148008, 2021.
- David Gries, *The Science of Programming*, Springer, 1981 (a classical text).

Unité d'enseignement 205.2 – *Beyond relational databases (BeyDB)*

Périodes/sem	4	Semestre	Printemps
Enseignant-e-s	Dr René Schumann, Dr Dimi Racordon	Coefficient	4

Description courte / objectifs

Le domaine de l'ingénierie des données couvre les étapes cruciales depuis l'acquisition des données brutes jusqu'à la mise à disposition des données validées et nettoyées en vue de leur exploitation. Ce cours a pour objectifs principaux d'explorer les paradigmes de gestion de base de données qui s'écartent du model relationnel classique, d'étudier leur mise en pratique et de comparer et contraster les méthodes actuelles et leur application. L'objectif principal visé est ainsi les méthodes de traitement de données non structurées.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Contrôle continu et examen semestriel, de poids respectifs 50%/50%.

Contenu (mots-clefs)

- Limites du modèle relationnel et modélisation de données
- Formatage et transformation des données : **JSON** (validation et schéma), **XML**, **YAML**, **parquet**, **VectorDB**, sérialisation/désérialisation
- Données non structurées vs données structurées
- NoSQL, document databases, graph databases, datalakes
- Séries temporelles
- **Databases management** : reliability, sauvegardes, versioning, déploiement et performances

Support de cours

- Documents mis à disposition par les professeur-e-s sous format numérique.
- Séries d'exercices ou laboratoires et ressources en ligne.

Unité d'enseignement 205.3 – Full-stack Web development (FStack)

Périodes/sem	4	Semestre	Printemps
Enseignant-e	Dr Guillaume Zufferey	Coefficient	4

Description courte / objectifs

Ce cours offre aux étudiant.e.s une immersion dans le monde du développement web full-stack, en abordant les thématiques essentielles de ce secteur désormais incontournable. Nous examinerons les défis associés à la gestion des données et à la conservation de l'état, tant du point de vue de la présentation (front-end) que de la gestion, du stockage et du traitement des données (back-end).

Ce cours privilégiera une approche pratique. Suivant le modèle de la classe inversée, les étudiant.e.s prépareront en amont chaque session en se plongeant dans la théorie et en réalisant des exercices fournis dans le support de cours. Les séances en classe seront principalement axées sur des ateliers où les concepts théoriques seront concrètement appliqués. Ainsi, les étudiant.e.s concevront progressivement une application web full-stack complète.

Nous explorerons de manière structurée les éléments constitutifs d'une application web. Notre parcours commencera côté client, en abordant les fondamentaux comme HTML, CSS et Javascript. Nous nous pencherons également sur divers frameworks, en soulignant leur pertinence et utilité. Le développement back-end suivra, en mettant l'accent sur des technologies comme Node.js. Notre exploration se poursuivra avec l'étude de la communication entre le back-end et le front-end via REST et GraphQL, ce qui nous éclairera sur la gestion des flux de données et les aspects cruciaux de la sécurité, tels que l'authentification et la gestion des droits d'accès. En conclusion, nous nous concentrerons sur le processus de déploiement d'une application web, en mettant l'accent sur l'automatisation (CI/CD) et en examinant diverses catégories de tests, tels que unitaires, fonctionnels ou end-to-end.

Objectifs

- Maîtriser les fondamentaux du développement *frontend* pour l'affichage et la manipulation des données
- Maîtriser les principes du développement backend pour le traitement et le stockage des données
- Comprendre les enjeux de la circulation et le traitement des données dans une application web
- Découvrir les techniques et outils pour la maintenance de l'état dans les interactions client-serveur
- Appliquer les compétences acquises pour développer et déployer une application web complète

Méthode(s) d'enseignement

- | | | |
|--|---|---|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☒ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre. Le cas échéant, le poids de chaque note est précisé en début de semestre. Cette unité d'enseignement ne comporte pas d'examen semestriel.

Contenu (mots-clefs)

- **Introduction**
 - Du site web statique à l'application web
 - Composants principaux (client, serveur, base de données, ...)
 - Communication client-serveur, protocoles HTTP et HTTPS
 - Principaux défis

- **Développement front-end**

- HTML: tags et structure, DOM, HTML5
- CSS: propriétés et valeurs, sélecteurs, design responsive et pré-processeurs
- Javascript: manipulation du DOM, gestion des événements, *promises*, *async/await*
- Frameworks ([React](#), [Vue](#), ...) : pourquoi et comment

- **Développement back-end**

- Framework [Node.js](#)
- Bases de données

- **Communication client-serveur**

- Serveur: APIs et frameworks (REST, GraphQL)
- Client: connexion et interaction avec le serveur (Ajax, fetch, ...)

- **Sécurité**

- Authentification et autorisation
- Gestion des sessions
- Sécurité des communications

- **Déploiement et hébergement**

- Stratégies de déploiement
- Intégration et déploiement continus (CI/CD)
- Tests automatisés
- Plateformes (par ex. [Netlify](#))

Support de cours

Support de cours numérique, références sur le web, exercices en ligne, laboratoires.

Outils utilisés

- Langages : [HTML](#), [CSS](#), [Javascript](#)
- Frameworks: [React](#), [Node.js](#), [GraphQL](#)
- Environnement de développement: choix libre ([Visual Studio Code](#) conseillé)

Bibliographie

- Douglas Crockford, *Javascript, the good parts*, ISBN- 97 805 96517748, 2008.

206 – Text and vision

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientation	Data engineering
Année académique	2025-2026		

1. Organisation du module

Module	Text and vision	Code du module	206
Type	Obligatoire	Crédits ECTS	11
Langue	Français, Anglais	Année académique	Deuxième année
Responsable(s)	Dr Louis Lettry	Organisation	Par discipline

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
206.1	DPVIS Data processing and visualization					4			
206.2	MDVIS Model driven machine vision					4			
206.3	FNL Formal and natural languages					3			

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	150 h	150 h	300 h

2. Description du module

Ce module a pour objectif d'approfondir les principes du traitement des données, notamment au travers de leur visualisation et de la vision par ordinateur, dans une double perspective :

- celle de la vision par ordinateur, dans laquelle des programmes interprètent et analysent des images issues du monde réel
- celle de la visualisation, qui nécessite de nettoyer, traiter et analyser des données afin de pouvoir les présenter sous forme visuelle

3. Prérequis

- ☒ Avoir validé le(s) module(s): 101
- ☒ Avoir suivi le(s) module(s): 200, 100
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

- analyser, évaluer et présenter visuellement un ensemble de donnée (**J**)
- transformer des données entre différentes représentations (**A**)
- concevoir et réaliser des systèmes de reconnaissance et reconstruction basés sur l'analyse de données (**A**)
- présenter visuellement des données de différents domaines (**A**)
- appliquer les principes de la vision par ordinateur (**C**)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation des différentes *unités d'enseignement* (UE) qui permettent le calcul de la **note du module** M comme suit :

$$M = \frac{4 \cdot m_{\text{DPVIS}} + 4 \cdot m_{\text{MDVis}} + 3 \cdot m_{\text{FNL}}}{11}$$

avec :

- m_{DPVIS} – moyenne des notes en *Data processing and visualization*;
- m_{MDVis} – moyenne des notes en *Model driven machine vision*;
- m_{FNL} – moyenne des notes en *Formal and natural languages*;

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

L'étudiant·e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant·e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint·e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

Unité d'enseignement 206.1 – Data processing and visualization (DPVIS)

Périodes/sem	4	Semestre	Printemps
Enseignant·e·s	Dr Vincent Andrearczyk, Dr René Schumann	Coefficient	4

Description courte / objectifs

Ce cours offre une introduction aux techniques de traitement et de visualisation des données. Il aborde des sujets tels que le nettoyage des données, le prétraitement, l'agrégation, le filtrage ainsi que la transformation des données. Vous y apprendrez les principes et les lignes directrices pour une visualisation efficace des données de différents types telles que les données multivariées, temporelles et géographiques ainsi que la visualisation de texte et de réseaux. Ces visualisations - interactives ou statiques - seront réalisées à l'aide des bibliothèques les plus utilisées dans le domaine.

Du point de vue théorique, vous apprendrez les techniques de réduction de dimensionnalité telles que PCA, t-SNE et UMPA ainsi que les différents algorithmes de regroupement pour la visualisation et la classification.

Ce cours sera accompagné de mini-projets afin de concevoir et présenter vos propres projets de traitement et de visualisation de données.

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre. Le cas échéant, le poids de chaque note est précisé en début de semestre. Cette unité d'enseignement ne comporte pas d'examen semestriel.

Contenu (mots-clefs)

- **Acquisition de données / exploration** : identification de tendances et extraction d'information
- **Nettoyage des données / prétraitement** : normalisation, standardisation, biais, *outliers*, imputation des données manquantes
- **Agrégation et analyse de données** : regroupement, résumé de données, calcul des mesures agrégées, filtrage
- **Visualisation de données numériques et alphanumériques** : graphiques, diagrammes et histogrammes (`matplotlib`, `d3.js`...)
- **Data-driven story telling** : récits et présentations de données efficaces, des données pour raconter des histoires et transmettre des informations
- **Éthique** : évaluation des visualisations de données, biais de visualisation, mentir avec des données

Support de cours

Slides, exercices, laboratoires et mini-projets.

Outils utilisés

- Langage `Python` et `Jupyter notebooks`
- Bibliothèques `matplotlib`, `D3`,...

Unité d'enseignement 206.2 – Model driven machine vision (MDV_is)

Périodes/sem	4	Semestre	Printemps
Enseignant·e	Dr Louis Lettry	Coefficient	4

Description courte / objectifs

Ce cours a pour but de présenter les principes de la vision par ordinateur guidée par modèle (*model driven*) avec une focalisation sur le domaine de la géométrie projective. Le cours commencera par présenter le fonctionnement d'un appareil photo/une caméra ainsi que les modèles mathématiques de projection les représentants. Le traitement d'image sera ensuite d'abord approché par la présentation d'opérations morphologiques denses, ayant lieu sur toute l'image, et ensuite d'opérations éparées sous la forme de points d'intérêts et leurs descriptions. Ces outils seront la base du début de la conception de modèle de vision qui utilisent le regroupement et l'appariement en lien avec différentes méthodes d'ajustement de modèles dans le domaine de la géométrie projective ainsi que de la détection/classification d'objets.

Méthode(s) d'enseignement

- | | | |
|--|---|---|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☒ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Acquisition** : acquisition et prétraitement d'un signal lumineux, modèle de projection et calibration, représentations de couleurs
- **Manipulation** : opérations morphologiques, gradient, contours
- **Points d'intérêt** : détection, description
- **Comparaison** : regroupement (clustering), appariement (matching), Hungarian algorithm
- **Ajustement de modèle** : transformée de Hough, RANSAC, J-Linkage,
- **Transformation** : homographie, triangulation, géométrie épipolaire, perspective-n-point, ajustement de faisceaux (bundle adjustment)
- **Reconstruction** : structure acquise à partir d'un mouvement (structure-from-motion), lumière structurée (structured light), vision stéréoscopique et carte de disparité
- **Détection** : sac-de-mots (bag of words), si le temps le permet *deformable-part-model*

Outils utilisés

- Librairie [OpenCV](#)
- Langage [Python](#) 3.x

Bibliographie

- R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd edition, 2011 - <https://szeliski.org/Book/>.

Unité d'enseignement 206.3 – *Formal and natural languages (FNL)*

Périodes/sem	3	Semestre	Printemps
Enseignant-e	Dr Renaud Richardet	Coefficient	3

Description courte / objectifs

Ce cours explore les fondements et les applications des langages formels réguliers et naturels, offrant une compréhension de la manière dont les langages informatiques et les langues humaines sont structurés, traités et analysés. À travers une approche pratique, ce cours examine les concepts clés des langages formels et réguliers (théorie des automates, langages et syntaxe formelle) ainsi que les spécificités des langues naturelles.

Concrètement, la partie formelle sera traitée au travers des expressions régulières puis au travers du développement d'un interpréteur de langage. La partie langage naturel couvrira des techniques de traitement du langage naturel telles que la correction orthographique, l'algorithme *tf-idf*, l'utilisation de moteurs de recherche, et les concepts de classification. Le cours abordera également de manière appliquée les dernières avancées technologiques telles que les modèles massifs de langues, la génération augmentée par récupération et leurs applications pratiques.}

Méthode(s) d'enseignement

- | | | |
|--|---|--|
| ☒ Cours et exercices | ☒ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☐ Mentoring |

Modalités d'évaluation

Au moins 1 note durant le semestre et 1 examen semestriel. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

1. Regular and formal languages

- Expressions régulières : méthodes pour rechercher des chaînes de caractères basées sur des motifs spécifiques. Théorie des automates et théorie des langages.
- Création d'un interpréteur pour analyser et exécuter un langage de programmation.

2. Natural Languages

- Introduction à la linguistique: fondamentaux de la science du langage, incluant phonétique, syntaxe et sémantique.
- Etude de l'évolution des systèmes de traduction automatique, des approches statistiques aux modèles basés sur l'intelligence artificielle.
- Correction d'orthographe: techniques et algorithmes pour identifier et corriger les fautes d'orthographe dans les textes.
- tf-idf* (Term Frequency-Inverse Document Frequency): mesure statistique utilisée pour évaluer l'importance d'un mot dans un document par rapport à une collection de documents ou corpus.
- Moteurs de recherche: technologies et algorithmes sous-jacents aux moteurs de recherche, incluant l'indexation et le ranking des résultats.
- Classification: méthodes pour identifier à quelle catégorie un document appartient basé sur ses caractéristiques.

3. Techniques avancées

- Transformer* - LLMs (*Large Language Models*): architecture de modèle basée sur des mécanismes d'attention, permettant de comprendre et de générer du texte.

- *Retrieval Augmented Generation* (RAG): technique combinant la génération de texte avec la recherche d'informations pour produire des réponses informées.
- Applications des modèles de langage à grande échelle dans divers domaines comme la traduction, le résumé, et la génération de texte.

Outils utilisés

- `spacy`: bibliothèque pour le traitement du langage naturel
- `scikit-learn`: bibliothèque pour le machine learning en Python qui fournit des outils simples et efficaces pour l'analyse de données textuelles
- `Elasticsearch`: application pour créer des moteurs de recherche
- `Huggingface transformer`: modèles massifs de langage

Bibliographie

- E. Kochmar, *Getting Started with Natural Language Processing: A friendly introduction using Python*, Manning, 2022
- A. Gupta, B. Majumber, S. Vajjala, *Practical Natural Language Processing*, O'Reilly, 2020
- L. von Werra, *Natural Language Processing with Transformers, Revised Edition*., O'Reilly, 2022.

207 – Summer school 2

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Intégration	Orientation	Data engineering
Année académique	2025-2026		

1. Organisation du module

Module	Summer school 2	Code du module	207
Type	Obligatoire	Crédits ECTS	3
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr M. Schumacher, Dr V. Andrearczyk	Organisation	Par bloc

2. Description du module

Le but de cette deuxième *summer school* est de réaliser un projet d'intégration permettant de mettre en pratique et action les connaissances acquises durant la seconde année. Pour cette *summer school*, la thématique abordée est celle du *développement mobile* dans le domaine des données.

Remarque : la présence aux activités de la *summer school* est obligatoire pour pouvoir valider le module.

3. Prérequis

- ☒ Avoir validé le(s) module(s): 201
- ☒ Avoir suivi le(s) module(s): 202, 205
- ☐ Pas de prérequis
- ☒ Autre : Si certains modules requis n'ont pas été acquis, sur discussion avec le responsable de filière.

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- spécifier des besoins d'une application (diagrammes *use cases*, mockups, *product backlog* et *user stories*) (A)
- mettre en place une gestion agile SCRUM sur un projet de groupe (A)
- appliquer les techniques du génie logiciel pour atteindre un code testé, de qualité, modulable et performant (A)
- développer l'agilité lors de la mise en oeuvre d'un *framework* de développement mobile (A)
- développer la pensée *ingénieur-e* par la réflexion individuelle et en groupe (A)
- sélectionner et appliquer les bonnes techniques et les bons algorithmes pour résoudre un problème dans un domaine spécifique (J)

5. Modalités d'évaluation et de validation

Il n'y a pas de note(s) dans ce module car il est évalué à l'aide d'une échelle *Acquis | Non acquis*.

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (C) Connaissances et compréhension, (A) Application, (J) Jugement (analyse, synthèse, évaluation).

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

Si le module n'est pas acquis, l'étudiant-e doit le refaire ou effectuer un complément (dans les cas limites discutés avec le/la RF).

301 – Machine learning and AI

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientations	Data engineering
Année académique	2025-2026		

1. Organisation du module

Module	Machine learning and AI	Code du module	301
Type	Obligatoire	Crédits ECTS	13
Langue	Français, Anglais	Année académique	Troisième année
Responsable(s)	Dr Louis Lettry	Organisation	Par bloc

2. Description du module

Ce cours vise à faire découvrir et appliquer les fondements du domaine de l'intelligence artificielle et plus particulièrement de l'apprentissage automatique par ordinateur.

Les étudiant·es seront amené·es progressivement à acquérir des compétences pratiques en matière de modélisation d'un problème d'apprentissage par ordinateur et de conception d'une solution, d'analyse des données d'entraînement et d'interprétation de résultats et modèles produits.

L'acquisition des compétences en concevant des solutions à des problèmes reposant sur l'analyse automatique de données. Ces multiples itérations permettront la découverte de diverses méthodes d'apprentissage tels que les *support vector machine*, *random forest*, *neural networks*. Les problèmes permettront également de montrer principaux principes et défis du domaine qui vont des différents types de supervisions pour l'apprentissage, les formulations standards telles que la régression ou la classification, etc.

Objectifs

L'objectif final est de former les étudiant·es à l'approche par apprentissage par ordinateur pour des problèmes de données réels et complexes et de leur faire acquérir les compétences relatives à la conception d'une telle solution et de ses différentes phases de l'acquisition et prétraitement des données en passant par la modélisation d'une solution ainsi que l'évaluation des modèles et résultats produits.

Contenus et mots clés

- **Prétraitement:** biais, augmentation de données, sélection de caractéristiques (feature selection), normalisation
- **Formalisation:** régression, classification, fonction d'erreur (loss function)
- **Supervision:** supervisé, non-supervisé, faiblement supervisé, apprentissage par imitation
- **Modèle d'apprentissage:** support vector machine (SVM), random forest, neural networks
- **Optimisation:** stochastique, descente de gradient
- **Évaluation:** métrique, ROC curve, precision, recall

Organisation

Ce module fera l'objet de quelques cours magistraux, de beaucoup de travaux pratiques ainsi que des travaux de projets. Les étudiant·es seront amené·es à travailler en groupe pour la réalisation de projets et à présenter leurs résultats à l'oral et

par écrit. Durant la 2ème partie du cours, les étudiant·es devront réaliser un projet complet de bout en bout, de la récolte des données à la présentation des résultats, en lien avec le module 302.

Outils utilisés

- Python, `Jupyter`, `scipy`, `scikit-learn`, autres en fonction du besoin.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 200, 201, 204, 205, 206
- ☐ Pas de prérequis
- ☒ Autre : Si certains modules requis n'ont pas été acquis, sur discussion avec le responsable de filière.

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

- Concevoir des solutions basée sur l'apprentissage par ordinateur (**J**)
- Analyser des données d'entraînement brutes et en extraire les tendances, identifier les potentiels biais, etc. (**J**)
- Préparer / pré-traiter / corriger des données d'entraînement pour les rendre adéquates pour un entraînement de modèle (**J**)
- Connaître différentes méthodes d'apprentissage par ordinateur et être capable de les comparer/commenter (**A**)
- Implémenter un pipeline d'apprentissage complet: récolte et préparation des données, entraînement du modèle, inférence basée sur le modèle entraîné (**J**)
- Commenter et critiquer un pipeline d'apprentissage par ordinateur, les modèles produits et les résultats d'inférence (**J**)

Compétences humaines

En plus des compétences techniques ci-dessus, les étudiant·es

- créent un climat de travail favorable en gérant les conflits de manière professionnelle si nécessaire
- valorisent les points forts des autres membres de l'équipe et ont confiance en leurs compétences
- sont capables de planifier et de vérifier un projet à l'aide d'outils logiciels professionnels.
- sont capables de définir leur propre rôle par rapport aux autres et d'assumer la responsabilité de leurs tâches
- peuvent appliquer les connaissances acquises dans les cours techniques au projet
- sont capables d'identifier les domaines dans lesquels ils et elles manquent de connaissances. Ils et elles sont capables d'organiser l'acquisition des connaissances nécessaires
- se soutiennent mutuellement dans l'acquisition des connaissances. C'est-à-dire qu'ils et elles apprennent les uns des autres et sont prêts à partager leur expertise
- sont capables de remettre en question les artefacts de leur projet (code, documents, procès-verbaux). Ils et elles vérifient et évaluent continuellement ces artefacts
- comprennent les résultats du travail des autres et les approches choisies par leurs coéquipiers
- sont capables de détecter systématiquement les erreurs et de fournir une solution potentielle
- connaissent les règles d'un retour d'information efficace. Ils et elles sont capables de fournir et d'accepter un retour d'information
- planifient et réalisent des tests d'acceptation avec les parties prenantes.

5. Modalités d'évaluation et de validation

Des évaluations hebdomadaires auront lieu lors de la première partie du cours. Différentes modalités seront utilisées en fonction de l'adéquation au sujet discuté. Durant cette partie, les étudiant·es seront évalué·es au moyen de QCMs,

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

exercices/laboratoires notés, présentations orales ainsi que rapports. L'évaluation de la deuxième partie du cours (projet commun) sera basée sur une présentation orale et un rapport écrit ainsi que sur la démonstration des performances de la solution développée. Les poids des différentes notes sera indiqué par les enseignant·es en début de semestre.

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant·e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant·e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant·e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint·e.

302 – Data computation

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientation	Data engineering
Année académique	2025-2026		

1. Organisation du module

Module	Data computation	Code du module	302
Type	Obligatoire	Crédits ECTS	13
Langue	Français, Anglais	Année académique	Troisième année
Responsable(s)	Dr Oscar Esteban	Organisation	Par bloc

2. Description du module

Les applications d'ingénierie des données, tels que l'analyse de données par l'apprentissage par ordinateur ou la visualisation, nécessitent bien souvent un déploiement et une exécution sur des infrastructures cloud. De plus, l'ingénierie des données requiert le stockage et la manipulation de grandes quantités de données qui doivent être accessibles de manière distribuée et de manière fiable. Finalement, certains calculs nécessaires à ce champ requièrent de déployer des solutions de calcul à haute performance, avec des méthodes et techniques assurant la reproductibilité et la scalabilité des solutions.

Ce module aborde ce sujet sous trois angles :

1. La mise en place d'infrastructures et de software stacks virtualisées, containerisées et orchestrées.
2. Le traitement de données de manière distribuée pour un approche *scale out*, adapté pour *big data* et applications data-intensive.
3. Le calcul à haute performance, efficacité et *scale up*, adapté pour applications compute-intensive.

L'objectif final est de former les étudiant·e·s à déployer des solutions de calcul et de stockage de données sur des infrastructures cloud et *on premise*, de manière fiable, reproductible et scalable.

Objectifs

Ce module permettra aux étudiant·es de pouvoir analyser (catégoriser, examiner et comparer) et choisir des outils liés à l'informatique cloud, le calcul distribué et à haute performance dans le contexte de l'ingénierie des données. Une attention particulière est accordée à une approche qui reflète des méthodes de travail utilisées dans le monde professionnel.

Contenus et mots clés

- **Virtualisation** : containerisation et environnements virtuels, construction et configuration d'un petit centre de calcul composé de machines virtuelles
- **Automatisation** : CI/CD, DevOps, MLOps, déploiement et configuration d'applications dans un cluster Kubernetes
- **Paradigmes de systèmes distribués** : HDFS et MapReduce, architecture Spark (RDDs et manipulation), dataframes et manipulations complexes multi-nœuds (Spark) et multi-threads (Polars)
- Real-world data processing and pipelines
- **Programmation parallèle et HPC** : MPI, OpenMP et CUDA, nœuds de calcul (nodes), cœurs physiques (cores) et cœurs logiques (threads), mémoire partagée et distribuée

Organisation

Ce module fera l'objet de quelques cours magistraux, de beaucoup de travaux pratiques ainsi que des travaux de projets. Les étudiant-es seront amené-es à travailler en groupe pour la réalisation de projets et à présenter leurs résultats à l'oral et par écrit. Durant la 2ème partie du cours, les étudiant-es devront réaliser un projet complet de bout en bout, de la récolte des données à la présentation des résultats, en lien avec le module 301.

Outils utilisés

- APIs et outils pour la création et configuration automatique de machines virtuelles ([OpenStack](#)/[CloudStack](#), [Terraform](#)/[Pulumi](#), [Packer](#))
- Docker files, docker compose, Kubernetes, Helm charts
- Polars, Spark, HDFS
- PyTorch, OpenMP, CUDA, MPI

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 201, 202
- ☐ Pas de prérequis
- ☒ Autre : Si certains modules requis n'ont pas été acquis, sur discussion avec le responsable de filière.

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- déployer, sécuriser et maintenir de manière automatique, reproductible et interopérable des services en production à haute disponibilité sur des serveurs cloud et locaux (**J**)
- comprendre et appliquer les mécanismes et techniques utilisés dans le contexte du cloud computing et des systèmes distribués afin de pouvoir les appliquer professionnellement (**A**)
- aborder un problème de calcul de données et utiliser des plateformes d'analyse de données distribuées à la pointe de la technologie (**A**)
- choisir les outils efficaces pour traiter des données de taille et de nature différentes (brutes, prétraitées) en tenant en compte des limites de l'infrastructure sous-jacente (**J**)
- discerner l'applicabilité de paradigmes de programmation distribués selon la complexité et la taille de jeu de donnée (**J**)
- travailler en équipe en utilisant une méthodologie agile (**J**)
- analyser la parallélisation d'un code en utilisant des méthodes appropriées (efficacité, speed-up, scaling) (**J**)
- paralléliser un code séquentiel efficacement par rapport aux contraintes de l'infrastructure et du projet (**A**)

Compétences humaines

En plus des compétences techniques ci-dessus, les étudiant-es

- créent un climat de travail favorable en gérant les conflits de manière professionnelle si nécessaire
- valorisent les points forts des autres membres de l'équipe et ont confiance en leurs compétences
- sont capables de planifier et de vérifier un projet à l'aide d'outils logiciels professionnels.
- sont capables de définir leur propre rôle par rapport aux autres et d'assumer la responsabilité de leurs tâches
- peuvent appliquer les connaissances acquises dans les cours techniques au projet
- sont capables d'identifier les domaines dans lesquels ils et elles manquent de connaissances. Ils et elles sont capables d'organiser l'acquisition des connaissances nécessaires

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

- se soutiennent mutuellement dans l'acquisition des connaissances. C'est-à-dire qu'ils et elles apprennent les uns des autres et sont prêts à partager leur expertise
- sont capables de remettre en question les artefacts de leur projet (code, documents, procès-verbaux). Ils et elles vérifient et évaluent continuellement ces artefacts
- comprennent les résultats du travail des autres et les approches choisies par leurs coéquipiers
- sont capables de détecter systématiquement les erreurs et de fournir une solution potentielle
- connaissent les règles d'un retour d'information efficace. Ils et elles sont capables de fournir et d'accepter un retour d'information
- planifient et réalisent des tests d'acceptation avec les parties prenantes.

5. Modalités d'évaluation et de validation

Des évaluations hebdomadaires auront lieu lors de la première partie du cours. Différentes modalités seront utilisées en fonction de l'adéquation au sujet discuté. Durant cette partie, les étudiant-es seront évalué-es au moyen de QCMs, exercices/laboratoires notés, présentations orales ainsi que rapports. L'évaluation de la deuxième partie du cours (projet commun) sera basée sur une présentation orale et un rapport écrit ainsi que sur la démonstration des performances de la solution développée. Les poids des différentes notes sera indiqué par les enseignant-es en début de semestre.

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant-e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

303 – Semester project

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Intégration	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Semester project	Code du module	303
Type	Obligatoire	Crédits ECTS	4
Langue	Français	Année académique	Troisième année
Responsable(s)	Dr Grégory Mermoud	Organisation	Par bloc

2. Description du module

Le projet de semestre consiste en un développement informatique concret et complet, réalisé de manière individuelle et différent pour chaque individu.

Il se base sur un cahier de charges défini par le ou la responsable du projet qui assure le suivi de celui-ci.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☐ Avoir suivi le(s) module(s)
- ☒ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- Analyser une problématique donnée et formaliser une solution adéquate
- Développer un projet informatique complet faisant usage de technologies connues ou à découvrir, en tenant compte des besoins et des risques sécuritaires
- Travailler de manière autonome et professionnelle en faisant usage des outils métiers dans le contexte d'un projet informatique
- Planifier, gérer, conduire et valoriser un projet informatique
- Communiquer de manière professionnelle et structurée sur un projet informatique

5. Modalités d'évaluation et de validation

Le projet de semestre est évalué par un rapport écrit et une présentation orale. Les critères d'évaluation sont définis dans le cahier des charges du projet.

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant-e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.

304 – Major option track

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientation	Data engineering
Année académique	2025-2026		

1. Organisation du module

Module	Major option track	Code du module	304
Type	Cours à choix	Crédits ECTS	10
Langue	Français, Anglais	Année académique	Troisième année
Responsable(s)	Dr Louis Lettry	Organisation	Par bloc

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
304.1	ROBOT	Robot control and generative AI						40	
304.2	MedDS	Medical informatics and data science						40	
304.3	Compiler	Modern compiler construction						40	
304.4	LLMS	LLMS and agents						40	

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
	100 h	200 h	300 h

2. Description du module

Les ingénieur·es en informatique doivent être capables - en plus de leurs connaissances techniques pointues dans leur domaine - de travailler dans différents domaines spécialisés de l'ingénierie. Ce module propose **différentes options d'approfondissement à choix** pour approfondir les connaissances dans un domaine spécifique à la croisée de différents domaines techniques de l'ingénierie.

Ces options d'approfondissement – liées avec les spécialités en recherche de la *Haute Ecole d'Ingénierie* – permettent aux étudiant·es de se spécialiser dans un domaine spécifique de l'informatique, tout en leur offrant une vue d'ensemble des différentes facettes des métiers de l'ingénierie.

La complexité des tâches demandées vise à mettre les étudiant·es dans des situations nécessitant la mobilisation de connaissances et compétences acquises lors de leurs études à tous les niveaux : aussi bien des domaines techniques et théoriques que des domaines de gestion de projet et de communication, ainsi que de l'adaptabilité et de l'apprentissage. La mobilisation des compétences variées dans un contexte pluridisciplinaire de découverte et de production forme l'apogée de la formation d'ingénieur·e en informatique et systèmes de communication.

Spécificité

Chaque étudiant·e choisit l'une des UE parmi celles proposées et suit les cours correspondants. Toutes les unités sont organisées en blocs.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 301, 302, 303
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

- Évaluer et exploiter les nouvelles technologies avec maîtrise dans le contexte de l'orientation choisie (**J**)
- Analyser et utiliser des technologies associées à des projets de recherche et développement de l'orientation (**A**)
- Analyser et résoudre des problèmes complexes en lien avec l'orientation (**A**)
- Communiquer efficacement et de manière professionnelle pour travailler en équipe dans le contexte d'un problème issu de situation professionnelles plausibles (**J**)

5. Modalités d'évaluation et de validation

Il n'y a pas de note(s) dans ce module car il est évalué à l'aide d'une échelle *Acquis* | *Non acquis*.

6. Modalités de remédiation

- ☐ Remédiation possible
- ☒ Pas de remédiation
- ☐ Autre

7. Modalités de répétition

Si le module n'est pas acquis, l'étudiant-e doit le refaire ou effectuer un complément (dans les cas limites discutés avec le/la RF).

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

Unité d'enseignement 304.1 – Robot control and generative AI (ROBOT)

Périodes/sem	40	Semestre	Printemps
Enseignant·e·s	Dr L. Lettry, Dr M. Richard, Loïc Azzalini	Coefficient	10

Description courte / objectifs

Ce module a pour objectif d'approfondir les compétences en traitement de données, en apprentissage automatique et en contrôle intelligent dans un contexte très proche de situations professionnelles réelles.

En groupe et organisés comme une petite équipe de recherche et développement, les étudiant·es abordent un problème ouvert et complexe qui nécessite la mobilisation de nombreuses compétences, incluant par exemple :

- **Traitement du langage naturel (NLP)** : Conversion de la parole en texte (speech-to-text) et analyse et interprétation d'instructions orales.
- **Vision par ordinateur** : reconnaissance et analyse des objets 3D, analyse et contrôle qualité.
- **Robotique et contrôle** : théorie du contrôle, programmation de robots, planification et coordination.
- **Apprentissage automatique** : utilisation de techniques d'apprentissage appropriées au sein d'un système complexe.
- **Intégration de systèmes** : combinaison des différentes technologies (NLP, vision, robotique) et développement d'interfaces utilisateur.
- **Gestion de projet et travail d'équipe** : planification et répartition des tâches, gestion de projet agile.
- **Communication et collaboration** : communication au sein de l'équipe, avec les supérieurs hiérarchiques et avec les clients.
- **Résolution de problèmes complexes** : analyse et décomposition du problème en sous-tâches, approche itérative et amélioration continue.
- **Développement logiciel** : production en groupe d'un logiciel robuste, fiable et efficace, en accord avec les meilleures pratiques professionnelles (DevSecOps).
- **Gestion des données** : collecte et prétraitement des données pour l'entraînement des modèles, stockage et gestion efficace des données du projet.
- **Éthique et sécurité** : considérations sur la sécurité dans l'utilisation des robots et réflexion sur les implications éthiques de l'automatisation.

Cette unité d'enseignement permet aux étudiant·es de développer des compétences techniques avancées tout en les préparant aux défis du monde professionnel, en mettant l'accent sur l'innovation, la collaboration et la résolution de problèmes complexes.

Méthode(s) d'enseignement

- | | | |
|---|--|---|
| ☐ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☒ Apprentissage par problème | ☒ Mentoring |

Modalités d'évaluation

L'évaluation de cette unité est réalisée à l'aide d'un portfolio de compétences.

Contenu (mots-clefs)

- **Traitement du langage naturel (NLP)** : Whisper, LLMs (p.ex., LLaMa3.1), inférence et fine-tuning, modèles open-source.
- **Vision par ordinateur** : reconnaissance d'objet, segmentation, photogrammétrie, stéréoscopie.
- **Robotique et contrôle** : programmation et contrôle, planification.

- **Apprentissage automatique** : apprentissage profond, arbres de décision, apprentissage par imitation, supervisé, auto-supervisé et non-supervisé.
- **Gestion de projet et travail d'équipe** : méthodes agile, itérations, Scrum, Kanban, ShapeUp.
- **Communication et collaboration** : Rapports techniques et whitepapers, présentations techniques et commerciales.
- **Résolution de problèmes complexes** : divide-and-conquer, design thinking, approches systémiques.
- **Développement logiciel** : revues de code, tests unitaires, tests d'intégration, tests système, DevSecOps.

Bibliographie

Particularité d'organisation

Ce module est organisé selon un apprentissage par problèmes et le déroulement de celui-ci se déroule par bloc, durant 7 semaines.

Unité d'enseignement 304.2 – Medical informatics and data science (MedDS)

Périodes/sem	40	Semestre	Printemps
Enseignant·e·s	Dr O. Esteban, Dr B. Franceschiello	Coefficient	10

Description courte / objectifs

Ce module vise à explorer les problématiques de l'informatique médicale, notamment le traitement et l'analyse des données issues de dispositifs médicaux, dans un contexte proche de situations professionnelles réelles.

En groupe et organisés comme une petite équipe de recherche et développement, les étudiant·es aborderont un problème ouvert et complexe nécessitant la mobilisation de compétences variées, incluant par exemple :

- **Traitement du signal** : analyse et interprétation d'un signal 1D dans l'espace et dans le domaine de Fourier.
- **Traitement d'image** : analyse des objets multi-dimensionnels contenant des informations liées à la santé du patient, analyse et interprétation des résultats.
- **Pipelines de données et intégration de systèmes** : orchestrer des solutions de flux de travail en combinaison des différents algorithmes afin de rationaliser le traitement des données et métadonnées biologiques et médicales, en intégrant des outils de traitement pour transformer et analyser les données.
- **DevOps** : utiliser de manière appropriée la conteneurisation et d'autres techniques pour déployer des solutions dans un environnement spécifique (par ex. un hôpital), en évaluant les besoins en ressources et les plateformes de choix (cloud, HPC, etc.).
- **Science des données et apprentissage automatique** : appliquer de façon appropriée l'apprentissage automatique, l'apprentissage profond, l'intelligence artificielle et des modèles statistiques pour extraire les informations nécessaires à partir des données, créer des visualisations efficaces et développement d'interfaces utilisateur afin de préparer des rapports afin de soutenir la prise de décision médicale.
- **Compréhension des dispositifs médicaux et des bases des données** : interprétation des données extraits, analyses, stockage sécurisé.
- **Gestion de projet et travail d'équipe** : planification et répartition des tâches, gestion de projet agile.
- **Communication et collaboration** : communication au sein de l'équipe, avec les supérieurs hiérarchiques et avec les clients.
- **Résolution de problèmes complexes** : analyse et décomposition du problème en sous-tâches, outils d'attribution des sous-tâches, utilisation des instruments pour planification, livrables et milestones.
- **Développement logiciel** : production en groupe d'un logiciel robuste, fiable et efficace.
- **Gestion des données** : stockage des données selon des pratiques qui respecte la confidentialité du patient et le travail avec une base de données large échelle, gestion efficace des données **Ethique et sécurité** : considérations sur la sécurité dans l'utilisation des données médicales et réflexion sur les implications éthiques de l'automatisation.

Cette unité d'enseignement permet aux étudiant·es de développer des compétences techniques avancées tout en les préparant aux défis du monde professionnel, en mettant l'accent sur l'innovation, la collaboration et la résolution de problèmes complexes.

Méthode(s) d'enseignement

- | | | |
|---|--|---|
| ☐ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☒ Apprentissage par problème | ☒ Mentoring |

Modalités d'évaluation

L'évaluation de cette unité est réalisée à l'aide d'un portfolio de compétences

Contenu (mots-clefs)

- **Traitement du signal multidimensionnel** : Matlab, Python, éléments du cours de *signal processing and information*.
 - **Informatique médicale** : tâches principales des dispositifs médicaux, traitement des données patient, confidentialité.
 - **Gestion de projet et travail d'équipe** : répartitions des tâches, travail d'équipe, outils pour attribution des tâches, Gantt charts, deliverables et milestones.
 - **Communication et collaboration** : rapports techniques et whitepapers, présentations techniques et commerciales.
- Résolution de problèmes complexes** : divide-and-conquer, design thinking, approches systémiques.
- **Développement logiciel** : revues de code, tests unitaires, tests d'intégration, tests système, integration aux dispositifs médicaux, DevOps.

Bibliographie

Particularité d'organisation

Ce module est organisé selon un apprentissage par problèmes et le déroulement de celui-ci se déroule par bloc, durant 7 semaines.

Unité d'enseignement 304.3 – Modern compiler construction (Compiler)

Périodes/sem	40	Semestre	Printemps
Enseignant·e	Vacat.	Coefficient	10

Spécificité

Ce cours **n'est pas proposé** durant l'année académique 2025-2026.

Unité d'enseignement 304.4 – LLMS and agents (LLMS)

Périodes/sem	40	Semestre	Printemps
Enseignant·e	Vacat.	Coefficient	10

Spécificité

Ce cours **n'est pas proposé** durant l'année académique 2025-2026.

305 – Minor option track

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Nouvelles technologies	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Minor option track	Code du module	305
Type	Cours à choix	Crédits ECTS	5
Langue	Anglais	Année académique	Troisième année
Responsable(s)	Dr. Davide Calvaresi	Organisation	Par bloc

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
305.1	IOT	IoT and distributed AI							40
305.2	CYBERSEC	Applied cybersecurity: defense and attack							40
305.3	RWDS	Real-world data science							40
305.4	CG	Computer graphics and games							40
305.5	Reverse	Reverse engineering							40
305.4	DEM	Digital energy management							40

TABLE 1 – Unités d'enseignement, en périodes hebdomadaires (45 min)

Volume de travail	Enseignement	Travail personnel	Total
40 h		110 h	150 h

2. Description du module

Les ingénieur-es des données sont amené-es à travailler dans différentes thématiques en fonction des besoins des entreprises dans lesquelles ils et elles sont engagé-es. Ce track de spécialisation permet de mettre l'accent – au travers de **différentes options d'approfondissement à choix** sur l'une de ces facettes en permettant un approfondissement des sciences des données et de son ingénierie. Chaque option est liée avec les spécialités en recherche de la *Haute Ecole d'Ingénierie* et leur donne une perspective de la recherche dans le domaine.

Spécificité

Chaque étudiant-e choisit l'une des UE parmi celles proposées et suit les cours correspondants. Toutes les unités sont organisées en blocs.

3. Prérequis

- ☐ Avoir validé le(s) module(s)
- ☒ Avoir suivi le(s) module(s): 301, 302, 303, 305
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant·e est capable¹ de :

- Évaluer et exploiter les nouvelles technologies avec maîtrise dans le contexte de l'orientation choisie (**J**)
- Analyser et utiliser des technologies associées à des projets de recherche et développement de l'orientation (**A**)
- Analyser et résoudre des problèmes complexes en lien avec l'orientation (**A**)
- Communiquer efficacement et de manière professionnelle pour travailler en équipe dans le contexte d'un problème issu de situation professionnelles plausibles (**J**)

5. Modalités d'évaluation et de validation

L'évaluation du module se base sur l'évaluation de l'une des *unités d'enseignement* (UE) à l'exception de toutes les autres. La **note du module** est la note de l'unité d'enseignement. Toutes les notes et moyennes sont précisées au dixième de point.

Conditions de réussite :

- Note du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant·e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant·e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant·e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint·e.

8. Contenus

Les descriptifs d'UE sont définis dans les pages suivantes.

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : (**C**) Connaissances et compréhension, (**A**) Application, (**J**) Jugement (analyse, synthèse, évaluation).

Unité d'enseignement 305.1 – IoT and distributed AI (IOT)

Périodes/sem	40	Semestre	Printemps
Enseignant·e·s	Christopher Métrailler, Dr Davide Calvaresi	Coefficient	5

Description courte / objectifs

Ce cours se penche sur la convergence de l'Internet des objets (IoT) et de l'intelligence artificielle distribuée (IA), en explorant comment les systèmes intelligents peuvent être intégrés dans l'environnement pour interagir, partager des données et prendre des décisions en temps réel. Les étudiant·es acquerront une expérience pratique avec des capteurs et des actionneurs intégrés, en apprenant à collecter et à analyser des données provenant du monde physique et virtuel.

Le cours est basé sur des projets. L'étudiant·e est initié·e aux concepts nécessaires et guidé dans la mise en œuvre du projet. Le résultat attendu est double : (i) l'acquisition de connaissances en pipeline complet (de la collecte de données au raisonnement distribué) et (ii) la mise en œuvre de la détection et du raisonnement distribués, où de multiples dispositifs communiquent (i.e., BLE) et collaborent pour atteindre des objectifs complexes. L'étudiant·e explorera les techniques de partage de l'information entre les dispositifs, la négociation des ressources et le raisonnement autonome pour optimiser les résultats dans un réseau distribué simple.

Méthode(s) d'enseignement

- | | | |
|--|---|---|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☒ Apprentissage par projet | ☐ Apprentissage par problème | ☒ Mentoring |

Modalités d'évaluation

Au moins 2 notes durant le déroulement du module. Le cas échéant, le poids de chaque note est précisé en début de semestre.

Contenu (mots-clefs)

- **Détection distribuée**: Lire les données de plusieurs capteurs intégrés pour une compréhension globale de l'environnement, en les partageant sur un réseau d'appareils pour des analyses plus précises.
- **Filtrage et stockage des données**: Nettoyer les données brutes et les stocker de manière distribuée pour une analyse et un raisonnement futurs.
- **Raisonnement sur les données**: Analyser les données pour identifier des tendances, prendre des décisions éclairées, et élaborer des plans en temps réel.
- **Agir sur l'environnement**: Utiliser des actionneurs (moteurs, DEL, etc.) pour ajuster l'environnement en réponse aux décisions prises.
- **Raisonnement distribué**: Coordonner des entités distribuées (capteurs, actionneurs) pour atteindre des objectifs communs grâce à des algorithmes d'IA.
- **Systèmes autonomes**: Développer des systèmes capables de fonctionner indépendamment avec une adaptation continue.
- **Réseaux IoT**: Gérer un réseau d'appareils interconnectés qui partagent des données et des commandes pour une collaboration fluide.
- **Systèmes cyber-physiques**: Concevoir des systèmes intégrant des algorithmes et des processus physiques pour un contrôle en temps réel.
- **Réseaux de capteurs sans fil**: Déployer des réseaux de capteurs communicant sans fil pour surveiller les conditions environnementales.

- **Systèmes multi-agents** : Créer des systèmes d'agents intelligents qui collaborent pour résoudre des problèmes complexes.
- **Négociation de l'information et des ressources**: Mettre en œuvre des protocoles permettant aux dispositifs de négocier et partager efficacement les ressources.
- **Décisions intelligentes** : Développer des algorithmes d'IA pour des décisions autonomes visant à optimiser les performances et répondre aux changements environnementaux.

Support de cours

Selon les indication des professeur-es

Particularité d'organisation

Ce module est organisé en bloc.

Unité d'enseignement 305.2 – Applied cybersecurity: defense and attack (CYBERSEC)

Périodes/sem	40	Semestre	Printemps
Enseignant-e	Dr J.-L. Beuchat, Dr G. Mermoud, Dr P. Junod	Coefficient	5

Description courte / objectifs

Le cours *Applied Cybersecurity* propose une approche par projet, dans laquelle les étudiants travaillent en groupe sur des problématiques concrètes de cybersécurité inspirées de cas réels rencontrés par des entreprises. Chaque groupe étudie un sujet distinct. L'objectif est de concevoir un *Minimum Viable Product* (MVP) : une solution fonctionnelle, mais simplifiée, permettant de valider les principales hypothèses techniques et organisationnelles liées au problème traité. Les thématiques abordées dans ce cours peuvent par exemple être liées au renforcement de la sécurité des systèmes d'exploitation, à la détection d'intrusions par apprentissage automatique, ou encore à la protection des données dans des environnements *cloud*.

En complément du travail de groupe, le cours intègre des activités de transfert de connaissances entre les équipes. Chaque étudiant doit être capable de présenter de manière synthétique les projets des autres groupes. Ce format favorise l'apprentissage collaboratif, le développement de compétences transversales et la capacité à communiquer des enjeux complexes de cybersécurité à différents niveaux d'expertise.

Méthode(s) d'enseignement

- | | | |
|--|--|---|
| ☒ Cours et exercices | ☐ Laboratoires / TP | ☐ Classe inversée |
| ☐ Apprentissage par projet | ☒ Apprentissage par problème | ☒ Mentoring |

Modalités d'évaluation

Présentation de projet en groupe (50%), examen oral individuel (50%) dans lequel l'étudiant-e décrit sa contribution individuelle.

Contenu (mots-clefs)

- **Cybersécurité.** Ensemble de pratiques, technologies et processus visant à protéger les systèmes informatiques, les réseaux et les données contre les attaques, les dommages ou les accès non autorisés.
- **Analyse de risques.** Identification, évaluation et hiérarchisation des risques liés à la sécurité de l'information, afin de définir des mesures de protection adaptées.
- **Security by design.** Approche consistant à intégrer des principes de sécurité dès la phase de conception d'un système ou d'une application, pour limiter les vulnérabilités dès le départ.
- **Privacy by design.** Principe selon lequel la protection de la vie privée et des données personnelles doit être intégrée dès la conception des systèmes et des processus.
- **SecDevOps.** Intégration des pratiques de sécurité dans les processus de développement et d'exploitation (DevOps), afin d'assurer une sécurité continue et automatisée tout au long du cycle de vie des applications.
- **Conformité et réglementation.** Respect des normes, lois et règlements en matière de sécurité de l'information, notamment le **RGPD** pour les échanges avec l'UE et la **nLPD** pour les traitements de données en Suisse.
- **Communication et vulgarisation.** Capacité à expliquer des concepts techniques de cybersécurité à des publics variés, y compris non spécialistes, dans un cadre professionnel.

Support de cours

Selon les indication des professeur-es

Particularité d'organisation

Ce cours est organisé en bloc.

Unité d'enseignement 305.3 – *Real-world data science (RWDS)*

Périodes/sem	40	Semestre	Printemps
Enseignant·e	Vacat.	Coefficient	5

Spécificité

Ce cours **n'est pas proposé** durant l'année académique 2025-2026.

Unité d'enseignement 305.4 – *Computer graphics and games (CG)*

Périodes/sem	40	Semestre	Printemps
Enseignant·e	Vacat.	Coefficient	5

Spécificité

Ce cours **n'est pas proposé** durant l'année académique 2025-2026.

Unité d'enseignement 305.5 – Reverse engineering (Reverse)

Périodes/sem	40	Semestre	Printemps
Enseignant·e	Vacat.	Coefficient	5

Spécificité

Ce cours **n'est pas proposé** durant l'année académique 2025-2026.

Unité d'enseignement 305.4 – Digital energy management (DEM)

Périodes/sem	40	Semestre	Printemps
Enseignant·e	Vacat.	Coefficient	5

Spécificité

Ce cours **n'est pas proposé** durant l'année académique 2025-2026.

330 – Bachelor thesis

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Intégration	Orientation	Tronc commun
Année académique	2025-2026		

1. Organisation du module

Module	Bachelor thesis	Code du module	330
Type	Obligatoire	Crédits ECTS	15
Langue	Français, Anglais	Année académique	Troisième année
Responsable(s)	Dr. Pierre-André Mudry	Organisation	Par bloc

2. Description du module

Dans le projet final de travail de bachelor, les étudiant-es appliquent les connaissances acquises au cours de leurs études dans le cadre d'un travail important sur un secteur d'activité actuel. Le résultat vise à démontrer l'acquisition des compétences professionnelles d'un-e ingénieur-e en informatique afin de pouvoir délivrer le diplôme.

En général, le travail est composé de :

- une analyse du problème basée sur les exigences des parties prenantes (entreprise, utilisateur final, etc.)
- une approche conceptuelle/théorique de la résolution du problème
- la mise en œuvre d'une solution
- une analyse des résultats
- une comparaison des résultats obtenus avec des travaux connexes, des systèmes existants ou des concepts théoriques
- un rapport documenté du travail, s'adressant aux professionnels de l'informatique. Il comprend au minimum le point de départ du projet, la méthodologie, les résultats et une analyse des résultats. Le rapport doit être créatif, structuré de manière logique et systématique et rédigé dans un anglais adéquat.
- la présentation, qui s'adresse à un public professionnel et qui contient l'essentiel du problème et de sa solution ainsi qu'une démonstration des résultats
- la défense de ses propres concepts, solutions théoriques et pratiques devant un public technique critique.

3. Prérequis

- ☒ Avoir validé le(s) module(s): Tous les autres modules du programme ISC
- ☐ Avoir suivi le(s) module(s)
- ☐ Pas de prérequis
- ☐ Autre :

4. Compétences visées par le module

À l'issue de ce module, l'étudiant-e est capable¹ de :

1. Les objectifs d'apprentissage sont classés selon les trois degrés de la taxonomie des objectifs selon les descripteurs de Dublin, à savoir, par ordre de profondeur croissante : **(C)** Connaissances et compréhension, **(A)** Application, **(J)** Jugement (analyse, synthèse, évaluation).

- analyser des problèmes ouverts en informatique, travailler de manière techniquement correcte et intégrer les résultats obtenus
- réfléchir sur sa propre méthodologie et examiner de manière critique les résultats obtenus
- décrire correctement et professionnellement tous les aspects essentiels du projet dans un rapport de projet
- formuler le rapport de projet de manière techniquement correcte et en s'appuyant sur des faits afin de rendre leurs résultats reproductibles.
- présenter avec assurance et compétence à un public professionnel les relations techniques, les méthodologies ainsi que les résultats envisagés de leur travail de projet
- appliquer de manière appropriée les techniques de gestion de projet et de documentation
- faire avancer leur projet de manière autonome en communiquant activement avec toutes les parties prenantes.

5. Modalités d'évaluation et de validation

Les modalités d'évaluation sont données par le responsable de filière en début de module. En cas d'échec au travail de bachelor, l'étudiant-e doit soit le refaire dans les délais fixés par le règlement d'études ou proposer un complément, en fonction de la décision du comité des travaux de bachelor.

Les notes et moyennes sont précisées au dixième de point, la note finale du module M est arrondie au demi-point.

Conditions de réussite :

- Note finale du module $M \geq 4.0$ (arrondie au demi-point).

6. Modalités de remédiation

- ☒ Remédiation possible
- ☐ Pas de remédiation
- ☐ Autre

La remédiation est possible uniquement lorsque celle-ci est prévue et que la note $M = 3.5$. Les modalités et le contenu de la remédiation sont fixés par le/la responsable de filière et communiqués à l'étudiant-e dès que possible dès lors qu'une remédiation est requise.

7. Modalités de répétition

L'étudiant-e qui répète un module ne refait pas les unités d'enseignement du module dont la moyenne m_i , arrondie au $\frac{1}{2}$ point, est égale ou supérieure à 5.0. En cas de répétition, l'étudiant-e peut sur demande refaire une unité d'enseignement à laquelle il/elle n'est pas astreint-e.