

PLAN D'ÉTUDES

Informatique et systèmes de communication (ISC)

2023–2024

Secrétariat académique
Médiatrice
Responsable de filière (RF)

Éliane Favre
Laurence Laffargue-Rieder
Dr Pierre-André Mudry

Arrêté le 5 février 2024
Version définitive

Table des matières

1.1	Introduction au plan d'études	4
1.2	Règlements de la filière	4
1.3	Programme de formation	5
1.4	100 - Sciences IT 1	7
1.5	101 - Fondements de la programmation 1	15
1.6	102 - Architecture matérielle	19
1.7	103 - Sécurité et communication	23
1.8	104 - Humanités 1	27
1.9	105 - Summer school 1	33
1.10	200 - Sciences IT 2	35
1.11	201 - Fondements de la programmation 2	39
1.12	202 - OS et DB	45
1.13	203 - Humanités 2	51
1.14	204 - Sciences IT 3	55
1.15	205 - Advanced programming	61
1.16	206 - Text and vision	69
1.17	207 - Summer school 2	75

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 droit à un certain nombre de crédits ECTS (*European Credit Transfer System*), tels que définis dans le modèle de Bologne. La formation bachelor comprend un total de 180 crédits qui sont répartis à raison de 60 crédits par année d'étude. Ils visent notamment à faciliter la mobilité estudiantine 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**, qui présente de manière synthétique la formation avec les modules qui la composent durant les trois années que dure la formation ;
- 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.

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 du module** – [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 *Informatique et systèmes de communication (ISC)*

Année académique 2023-2024

	Module	ECTS	Unités d'enseignement Cours	Poids	Pér. SA	Pér. SP
1 ^{ère} 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)	
			100.9 – Complément sciences IT (opt.)			(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)	
			101.9 – Complément fondements prog. 2 (opt.)			(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)	
	103 – Sécurité et communication	6	103.1 – Réseaux IP	3	4	
			103.2 – Cryptographie et sécurité	3		4
2 ^{ème} année, 60 ECTS	104 – Humanités 1	9	104.1 – Anglais 1	2	2	
			104.2 – Anglais 2	2		2
			104.3 – Gestion de projet IT	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	8	201.1 – Algorithmes et structures de données	4	4	
			201.2 – Génie logiciel	4	4	
	202 – OS, concurrence et DB	10	202.1 – Bases de données relationnelles	4	4	
			202.2 – OS et concurrence	6	6	
	203 – Humanités 2	4	203.1 – Éthique et droit IT	2	2	
			203.2 – Anglais 3	2	2	
	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
	207 – Summer school 2	3	207.1 – Mobile development summer school	3		Summer school 2
3 ^{ème} année, 60 ECTS	301 – Machine learning and AI	13	301.1 – Optimization, ML and deep learning	13	Bloc	
	302 – Data computation	14	302.1 – Big data, cloud and infrastructures	14	Bloc	
	303 – Semester project	3	303.1 – Hands-on development project	3	Bloc	
	304 – Major option track	10	304.1 – IoT and distributed AI	10		A choix
			304.2 – Robot control with learning	10		
			304.3 – Text and language in CS	10		
			304.4 – Digital energy management	10		
	305 – Minor option track	5	305.1 – Health and data	5		A choix
			305.2 – Computer graphics and games	5		
			305.3 – Real-world data science	5		
	330 – Bachelor thesis	15	330.1 – Travail individuel de bachelor	15		Bloc

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	2023-2024		

1. Organisation du module

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

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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** comme suit :

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

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*;

Toutes les notes et moyennes sont précisées au dixième de 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

Les modalités et le contenu de la remédiation sont fixés par le 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.

Unité d'enseignement 100.1 – Algèbre linéaire 1 (ALi n1)

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 d'enseignement

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

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 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 100.2 – Algèbre linéaire 2 (AL i n2)

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 d'enseignement

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

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 d'enseignement

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

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 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 100.4 – 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 d'enseignement

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

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 d'enseignement

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

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	2023-2024		

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		

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 logiciel. 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 dans une optique 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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** 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*;

Toutes les notes et moyennes sont précisées au dixième de 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 101.1 – Programmation impérative (PImp)

Périodes/sem	6	Semestre	Automne
Enseignant·e	Dr P.-A. Mudry	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 d'enseignement

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

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. **Etat** : variables et valeurs, assignation, types de base et conversions, opérateurs et expressions
2. **Flux de contrôle** : séquence d'instructions, instructions conditionnelles, répétitions
3. **Fonctions** : passage de paramètres, valeurs de retour, récursion simple;
4. **Manipulation de chaînes de caractères** : `String`, méthodes et interpolation
5. **Programming in practice** : outils de *debug* et de travail en groupe (`git`), flux de données, grammaires, `terminal`
6. **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	Dr P.-A. Mudry	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 d'enseignement

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

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)

- Orienté-objet** : classes, objets et propriétés, héritage, interfaces et délégation, classes données
- Typage** : généricité et polymorphisme, égalité structurelle et référentielle, structures de données complexes et abstraites (`List`, `Map`), sérialisation, réflexion (si le temps le permet)
- 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	2023-2024		

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		

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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** comme suit :

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

avec :

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

Toutes les notes et moyennes sont précisées au dixième de 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	Silvan Zahno	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 d'enseignement

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

Modalités d'évaluation

Un examen intermédiaire ainsi que la notation du projet de semestre en plus de l'examen semestriel. Le poids de chaque note est précisé en début de semestre.

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](#), [AMD Xilinx ISE/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 d'enseignement

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

Modalités d'évaluation

Au moins 2 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

Support de cours

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

Outils utilisés

- Langage de description du matériel VHDL
- [SiFive Freedom SDK](#)
- Environnement de développement [Mentor HDL-Designer](#), [Mentor Modelsim](#), [AMD Xilinx 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.

Particularité d'organisation

Pas d'heures spécifiques attribuées au laboratoire, les laboratoires sont définis de semaine en semaine.

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	2023-2024		

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		

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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** comme suit :

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

avec :

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

Toutes les notes et moyennes sont précisées au dixième de 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	Dr Gianluca Rizzo	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 :

- expliquer le modèle OSI ainsi que les différentes architectures de réseaux informatiques
- réaliser des mesures et des analyses de réseaux locaux
- classer et connaître les principaux algorithmes de routage
- utiliser les protocoles de communication de la couche liaison de donnée ainsi que le système d'adressage

Méthode d'enseignement

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

Modalités d'évaluation

2 notes durant le semestre et 1 examen semestriel. Le poids pour chaque note est le même.

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, Telnet, HTTP, proxies
- **Modèle de programmation** : `sockets` et communication client-serveur

Support de cours

Slides, laboratoires et mini-projets.

Outils utilisés

- Outils divers [Wireshark](#), [curl](#), [netcat](#), machines virtuelles [SLiTaz](#) et [OpenWRT](#).

Bibliographie

- J. Kurose, K. Ross, *Computer Networking: A Top-Down Approach*, 8th edition, ISBN-13 978-1292405469, 2021.

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

Périodes/sem	4	Semestre	Printemps
Enseignant·e·s	Dr Louis Lettry, 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 d'enseignement

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

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-clés)

- **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	2023-2024		

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		

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
104.1 Ang1	Anglais 1	2							
104.2 Ang2	Anglais 2		2						
104.3 GesProIT	Gestion de projet IT		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) ainsi qu'en gestion de projet.

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)**

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

- 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)**
- connaître les étapes d'un projet et appliquer les méthodes pour le gérer **(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** comme suit :

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

avec :

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

Toutes les notes et moyennes sont précisées au dixième de 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 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	2	Semestre	Automne
Enseignant·e	Erika Fauchère	Coefficient	2

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 du niveau B2 afin que les étudiant·e·s puissent commencer le cours de préparation au niveau C1 lors du semestre suivant.

Méthode d'enseignement

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

Modalités d'évaluation

Contrôle continu (2 notes) et examen de semestre oral. Le contrôle continu comprend des exposés, des épreuves orales ou écrites. Chacune des trois épreuves a le même poids.

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

L'objectif de ce cours est la préparation afin d'acquérir un niveau d'anglais suffisant pour pouvoir correspondre au niveau du certificat *Cambridge C1 Advanced*. A la fin de ce cours, les étudiant·e·s seront partiellement capables de :

- suivre un cursus académique au niveau universitaire
- 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 un haut niveau d'aisance

Méthode d'enseignement

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

Modalités d'évaluation

Contrôle continu (2 notes) et examen de semestre oral. Le contrôle continu comprend des exposés, des épreuves orales ou écrites. Chacune des trois 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

- H. Chilton, L. Edwards, *Formula C1 advanced coursebook without key & Ebook*, Pearson, ISBN-13 978-1-292-39149-6, 2021.

Unité d'enseignement 104.3 – Gestion de projet IT (GesProIT)

Périodes/sem	2	Semestre	Printemps
Enseignant·e	Frédéric Revaz	Coefficient	2

Description courte / objectifs

Ce cours a pour but de préparer les étudiant·e·s à la gestion de projet, notamment dans le contexte du développement informatique. Dans un premier temps, la problématique et les outils seront présentés. Ensuite, le cours sera orienté sur le plus d'applications pratiques possibles. À la fin du cours, l'étudiant·e sera capable :

- d'expliquer les enjeux de la gestion de projet
- d'expliquer comment utiliser les outils de la gestion de projet
- d'organiser, préparer et présenter un projet
- d'utiliser correctement les outils de gestion dans le cadre d'un vrai projet
- de communiquer correctement lors de séances de projet

Méthode d'enseignement

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

Modalités d'évaluation

Selon les instructions de l'enseignant·e.

Contenu (mots-clefs)

1. Méthodes et outils de gestion de projet :

- découpage en phases
- analyse de risque
- organisation
- planification
- budget
- qualité
- conduite de séances et PV

2. Application de méthodes de gestion de projet :

- dans différents cas d'utilisation en relation avec d'autres modules
- planification simple et personnelle
- planification et suivi d'un projet de développement informatique (notamment les méthodes dites *agiles*)

Particularité d'organisation

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

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

Périodes/sem	3	Semestre	Automne
Enseignant·e	Laurence Laffargue-Rieder	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 d'enseignement

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

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
- convaincre
- networking et small talk
- négocier et pitcher

3. Ateliers

- partager autour de la fresque du climat
- être à l'aise à l'oral, maîtrise de la voix et des émotions
- présenter avec impact et théâtraliser ses contenus
- communiquer par la vidéo

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	2023-2024		

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		

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 :

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

- utiliser des libraires pour analyser des données disponibles et concevoir des modèles (C)
- 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 proposer un complément (dans les cas limites).

200 – Sciences IT 2

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Sciences de l'ingénierie	Orientation	Tronc commun
Année académique	2023-2024		

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		

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 *Modélisation de phénomènes physiques*) 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)
- ☐ 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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** 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*;

Toutes les notes et moyennes sont précisées au dixième de 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 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 d'enseignement

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

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)

- **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 Gilles Evéquoz, 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 d'enseignement

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

Modalités d'évaluation

Une note durant le semestre (sous forme de mini-projet) et un examen semestriel. Le poids de chaque note est précisé en début de 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, 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	2023-2024		

1. Organisation du module

Module	Fondements de la programmation 2	Code du module	201
Type	Obligatoire	Crédits ECTS	8
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr Louis Lettry		

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			

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

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

2. Description du module

Le design de tout logiciel en ingénierie informatique procède de trois 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 dernière étape, à l'interface entre ces deux 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**)

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** comme suit :

$$M = \frac{4 \cdot m_{\text{Algo}} + 4 \cdot m_{\text{SoftEng}}}{8}$$

avec :

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

Toutes les notes et moyennes sont précisées au dixième de 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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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. Le projet permettra également d'aborder le domaine de l'apprentissage par ordinateur en découvrant l'optimisation guidée par les données.

Méthode d'enseignement

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

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 d'enseignement

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

Modalités d'évaluation

1 note durant le semestre et 1 examen semestriel oral lié à un mini-projet. 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
- Combinaisons de patterns
- Bad Smells et refactoring

4. Gestion de la qualité du logiciel

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

5. 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.

202 – OS, concurrence et DB

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Base des TIC	Orientation	Tronc commun
Année académique	2023-2024		

1. Organisation du module

Module	OS, concurrence et DB	Code du module	202
Type	Obligatoire	Crédits ECTS	10
Langue	Français	Année académique	Deuxième année
Responsable(s)	Dr Pamela Delgado		

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
202.1 DBRe1	Bases de données relationnelles				4				
202.2 OSCon	OS et concurrence				6				

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

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 C) mais également grâce aux différents outils de la programmation concurrente (en Java).

Dans le même temps, vous découvrirez comment sont organisés les systèmes de fichiers au niveau système et comment le principe des bases de données relationnelles (via la langage SQL) permet de structurer l'information de manière pratique et élégante en faisant un lien entre les données brutes et les données structurées.

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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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)
- 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** comme suit :

$$M = \frac{4 \cdot m_{\text{DBRel}} + 6 \cdot m_{\text{OSCon}}}{10}$$

avec :

- m_{DBRel} – moyenne des notes en *Bases de données relationnelles*;
- m_{OSCon} – moyenne des notes en *OS et concurrence*;

Toutes les notes et moyennes sont précisées au dixième de 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 – Bases de données relationnelles (DBRe1)

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

Description courte / objectifs

En informatique, 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 d'enseignement

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

Modalités d'évaluation

1 évaluation durant le semestre, 1 projet noté et 1 examen semestriel. 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.

Unité d'enseignement 202.2 – OS et concurrence (OSCon)

Périodes/sem	6	Semestre	Automne
Enseignant·e·s	Dr Pamela Delgado, Dr Michael Schumacher	Coefficient	6

Description courte / objectifs

Les systèmes d'exploitation sont une partie essentielle de tout système informatique car ils permettent d'abstraire le matériel et simplifier l'usage des ressources computationnelles (mémoire et CPU) à des programmes fonctionnant en parallèle. Les systèmes d'exploitation permettent ainsi d'exécuter des programmes, offrent des primitives pour la gestion de l'allocation du ou des processeurs, de la mémoire et des périphériques pour le stockage et la communication.

L'objectif de ce cours est de présenter des concepts des systèmes d'exploitation avec une approche pratique. Les sujets abordés comprennent l'organisation des systèmes d'exploitation, la programmation système, la gestion de la mémoire vive et les systèmes de stockage.

Le cours présente également 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, ...) pour gérer les problèmes qui surgissent avec la programmation concurrente.

A la fin du cours, l'étudiant·e sera capable de décrire les principes utilisés dans la conception des systèmes d'exploitation modernes et d'expliquer certaines des caractéristiques de base nécessaires pour supporter un tel système. L'étudiant·e sera également capable de comprendre et de fournir des solutions aux principaux défis qui se posent lors de la conception et de la mise en œuvre de systèmes concurrents.

Objectifs :

- être capable d'utiliser les concepts clés des systèmes d'exploitation : mémoire, gestion des tâches et scheduling, threads et processus, pipes;
- utiliser les outils d'un système d'exploitation pour traiter les données et automatiser les processus
- manipuler les primitives de la programmation système pour la gestion de la mémoire et des processus en langage C
- expliquer comment la gestion de la mémoire et du stockage fonctionne dans un système d'exploitation
- interpréter la virtualisation des ressources
- situer le domaine de programmation concurrente parmi les autres techniques de programmation
- comprendre les méthodes efficaces de structuration des programmes concurrents

Méthode d'enseignement

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

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-clés)

- **Programmation système** : C et *bash*
- **Gestion de la mémoire** : gestion, swapping, pagination, segmentation

- **Allocation des ressources** : planification et ordonnancement
- **Persistence** : disques, RAID, système de fichiers, E/S
- **Virtualisation**
- **Systèmes multiprocesseurs** : processus, communication interprocessus et synchronisation
- **Concurrence** : shared state et exclusion mutuelle, sémaphores, deadlocks, variables de condition
- **Programmation concurrente** : 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](#)
- Environnement Linux 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.
- [xv6: a simple, Unix-like teaching operating system](#).
- [Oracle The Java Tutorial - Lesson: Concurrency](#).
- Schild, H., *Java: The Complete Reference*, Twelfth Edition, ISBN 9781260463422, 2021, [online resource](#).

Particularité d'organisation

Le cours est réparti avec 2 heures de théorie et 4 heures de pratique en laboratoire par semaine.

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	2023-2024		

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		

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
203.1	EthIT	Éthique et droit IT			2				
203.2	Ang3	Anglais 3			2				

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 anglaise ainsi que sur les questions d'éthique et de droit en lien avec le domaine informatique.

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**)
- 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**)

1. Les objectifs d'apprentissage de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** comme suit :

$$M = \frac{2 \cdot m_{\text{EthIT}} + 2 \cdot m_{\text{Ang3}}}{4}$$

avec :

- m_{EthIT} – moyenne des notes en *Éthique et droit IT*;
- m_{Ang3} – moyenne des notes en *Anglais 3*;

Toutes les notes et moyennes sont précisées au dixième de 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 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 d'enseignement

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

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 203.2 – Anglais 3 (Ang3)

Périodes/sem	2	Semestre	Automne
Enseignant·e	Dana Mumford Jacquier	Coefficient	2

Description courte / objectifs

L'objectif de ce cours est d'acquérir un niveau d'anglais suffisant afin de pouvoir correspondre au niveau du certificat *Cambridge C1 Advanced*. A la fin de ce cours, les étudiants seront capables de :

- suivre un cursus académique au niveau universitaire
- 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 un haut niveau d'aisance

Méthode d'enseignement

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

Modalités d'évaluation

Le contrôle continu comprend des exposés, des épreuves orales ou écrites. Les poids des différentes épreuves est donné par l'enseignant·e en début de semestre.

Contenu (mots-clefs)

Différentes activités permettant aux étudiants 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

H. Chilton, L. Edwards, *Formula C1 advanced coursebook without key & Ebook*, Pearson, ISBN-13 978-1-292-39149-6, 2021

Particularité d'organisation

Cette unité d'enseignement constitue également une préparation au certificat *C1 Advanced* qui peut être également entrepris à la fin de l'UE.

204 – Sciences IT 3

Descriptif de module

Filière	Informatique et systèmes de communication	Formation	Bachelor
Axe d'enseignement	Orientation	Orientation	Data engineering
Année académique	2023-2024		

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		

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** 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*;

Toutes les notes et moyennes sont précisées au dixième de 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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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, et les réseaux. 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 d'enseignement

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

Modalités d'évaluation

Deux contrôles continus durant le semestre.

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)
- **Bruit** : détection de la présence de bruit et d'artefacts
- **Systèmes sur réseaux et réseaux d'ordre supérieur**

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

Langages [Julia](#) et [Octave](#).

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 utilisé 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 à l'analyse des réseaux).

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 d'enseignement

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

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)

- **Mécanique lagrangienne et optimisation** : équation d'Euler-Lagrange, optimisation sous contraintes (*linear programming, mixed integer linear programming*, méthodes heuristiques), optimisation pour la gestion (*dynamic programming, model predictive control*)
- **Mécanique statistique et processus stochastiques** : théorie cinétique des gaz parfaits (variables macroscopiques et microscopiques), processus stochastique (marche aléatoire, diffusion, réversibilité, entropie), méthode de Monte Carlo

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	Orientation	Data engineering
Année académique	2023-2024		

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		

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)

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

- juger les avantages et inconvénients du typage dans le contexte de la programmation et de la structuration des données (**J**)
- 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** 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*;

Toutes les notes et moyennes sont précisées au dixième de 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 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 205.1 – Functional programming (FunProg)

Périodes/sem	3	Semestre	Printemps
Enseignant·e·s	Dr P.-A. Mudry, Dr Renaud Richardet	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 d'enseignement

- ☒ Cours et exercices
- ☒ Laboratoires / TP
- ☒ Classe inversée
- ☒ Apprentissage par projet

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)

- **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 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.
- **Projets applicatifs**
 - Thématiques *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).

Particularité d'organisation

Le cours est donné à deux enseignant·e·s, l'un pour la partie théorique et le second pour la partie apprentissage par projet.

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

Périodes/sem	4	Semestre	Printemps
Enseignant·e·s	Dr René Schumann, Dr Pamela Delgado	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 d'enseignement

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

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)

- Limites du modèle relationnel
- Formatage et transformation des données : **JSON** (validation et schéma), **XML**, **YAML**, sérialisation/désérialisation
- Données non structurées vs données structurées
- NoSQL, document databases, graph databases, datalakes
- Séries temporelles
- 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 d'enseignement

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

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)

- **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	2023-2024		

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		

Code	Nom de l'unité	S1	S2	SS1	S3	S4	SS2	S5	S6
206.1	DPVIZ Data processing and visualization					4			
206.2	MDV is 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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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** comme suit :

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

avec :

- m_{DPVIZ} – 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*;

Toutes les notes et moyennes sont précisées au dixième de 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 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 (DPVIZ)

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 UMAP 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 d'enseignement

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

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)

- **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 `Javascript`
- Bibliothèques `matplotlib`, `D3.js`, ...

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

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 d'enseignement

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

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 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 d'enseignement

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

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	2023-2024		

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		

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 de ce module sont classés selon les trois degrés simplifiés de la taxonomie des objectifs d'apprentissage 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 proposer un complément (dans les cas limites).