CentraleSupélec

Course catalogue

Computer Science Engineer

Second Year

Metz Campus of CentraleSupélec

last update: January 25, 2026

Semester 7

ISP-INF-S07-05		**Artificial Intelligence S07**	**8 ECTS**
3MD1530	2.5	Statistical Models 1	29.0 h
3MD1540	3	Machine learning 1	35.5 h
3MD4040	2.5	Deep learning	28.5 h

ISP-INF-S07-06		**Computer Science 1 S07**	**7 ECTS**
SPM-INF-012	2	HPC on a single computing node	22.5 h
3MD4130	2	Big Data computation models	21.0 h
SPM-INF-011	3	Software engineering	36.0 h

ISP-INF-S07-27		**Computer Science 2 S07**	**7 ECTS**
SPM-INF-014	2	Langage processing	21.0 h
SPM-INF-013	2	Operations research	19.5 h
SPM-MAT-004	2	Optimization	25.0 h
SPM-MAT-007	1	Information Theory	11.0 h

ISP-INF-S07-23		**HEP S07**	**4 ECTS**
SPM-HEP-014	2	Business Management and Transformation	26.5 h
SPM-HEP-013	P/F	Conferences	10.0 h
SPM-HEP-019	1	Human ressources Management	18.0 h
SportS07	P/F	Sport S07	21.0 h

ISP-INF-S07-17		**Foreign Language S07**	**4 ECTS**
LV1S07	1	Foreign Languages and Culture 1	21.0 h
LV2S07	1	Foreign Languages and Culture 2	21.0 h

Semester 8

ISP-INF-S08-07		**Software Developpement S08**	**10 ECTS**
SPM-INF-015	3	Application Architectures	41.0 h
SPM-PRJ-003	7	Software development project	84.0 h

ISP-INF-S08-08		**Computer Science S08**	**9 ECTS**
SPM-INF-004	2.5	Introduction to Quantum Computing	35.5 h
SPM-INF-016	2.5	Programming paradigms	28.5 h
SPM-INF-017	2.5	Autonomous robotics	41.0 h
SPM-PRJ-004	1.5	Robotic and AI project	21.0 h

ISP-INF-S08-24		**HEP S08**	**7 ECTS**
SPM-HEP-011	1	Engineer, environment and society	14.0 h
SPM-HEP-018	1	Controversy	18.0 h
SPM-HEP-017	2	Systems engineering	21.5 h
SportS08	P/F	Sport S08	21.0 h
SPM-STA-002	4	Engineering internship	0.0 h

ISP-INF-S08-18		**Foreign Language S08**	**4 ECTS**
LV1S08	1	Foreign Languages and Culture 1	21.0 h
LV2S08	1	Foreign Languages and Culture 2	21.0 h

Statistical Models 1

Course supervisor: Frédéric Pennerath

Total: 29.0 h **CM:** 16.5 h, **TD:** 4.5 h, **TP:** 6.0 h

3MD1530 *back*

Description: The "Statistical Modeling" courses ModStat1 and ModStat2 deal with the modeling of systems for which the outputs are sufficiently uncertain that they need to be modeled by random variables. The course begins with a review of statistics and the introduction of elementary models (e.g. naive Bayes, linear regression, etc.), moving progressively towards more complex models. While the courses present the most useful elementary models and methods in this modeling context, they are not intended to be an exhaustive catalog. The aim is rather to present, within a consistent theory, the concepts and tools common to all these models and methods, and to show how, starting from modeling hypotheses specific to each concrete problem, these concepts are logically assembled before leading to an operational method. From a practical point of view, the aim of this course is not only to give students the means to understand and make good use of existing model implementations, but also to design their own implementations to take into account the specificities of a given problem. The course focuses on linking theory to practice: first, the hypotheses associated with a given class of problems are identified in class, followed by theoretical modeling work, leading to the definition of a model and its estimation algorithms. These results are then applied to a case study in tutorial sessions, before being implemented (in Python) and evaluated on data in practical exercises. The ModStat1 course will introduce the basic tools of statistical modeling, while the ModStat2 course will focus on hidden variable models.

Prerequisites: - Basic knowledge of probability theory, statistics and machine learning - Beginner level in Python / Numpy programming

Learning outcomes: - Be able to choose a statistical model/method adapted to the problem under consideration and implement it appropriately - Be able to understand the theoretical concepts underlying a statistical inference method presented in a scientific article. - Be able to implement a model / statistical method in a language such as Python. - Be able to adapt a model/method to take into account the specificities of the problem being addressed.

Evaluation methods: 2h written test without documents, can be retaken.

Evaluated skills:

- Modelling
- Research and Development

CM:

1. Modèles statistiques (1.5 h)
2. Estimation (1.5 h)
3. Estimation bayésienne (1.5 h)
4. Réseaux bayésiens 1 (1.5 h)
5. Naive Bayes (1.5 h)
6. Réseaux bayésiens 2 (1.5 h)
7. Causalité (1.5 h)
8. Modèles gaussiens (1.5 h)
9. Modèles linéaires (1.5 h)
10. Famille exponentielle (1.5 h)
11. GLM (1.5 h)

TD:

1. Estimation bayésienne (1.5 h)

2. Modélisation causale (1.5 h)
3. Régression (1.5 h)

TP:

1. Modélisation et modèles gaussiens (3.0 h)
2. Régression (3.0 h)

Machine learning 1

Course supervisor: Hervé Frezza-Buet

Total: 35.5 h **CM:** 13.5 h, **TP:** 20.0 h

3MD1540 *back*

Description: This course sets out the general framework of machine learning, allowing you to situate the different approaches in the field. It covers the notions of data pre-processing, an introduction to statistical learning theory (risks, overlearning, convex proxies, regularization), the difference between frequentist and Bayesian approaches, supervised, unsupervised, semi-supervised and reinforcement learning paradigms. Some approaches are detailed (Kernel methods, SVM, Boosting, Bagging, Decision trees...).

Learning outcomes: At the end of this course, students will be able to recognize the different classes of algorithms in the landscape of the many methods available on the shelf. They will also have the statistical notions that will enable them to make reasoned use of these methods, thus avoiding a black-box approach with blind parameter testing.

Means: The courses and practical work are given by Hervé Frezza-Buet, Arthur Hoarau, Jérémy Fix. The courses present theoretical aspects, mathematical proofs, but are also illustrated by demonstrations of algorithms. The practical work will be done in Python, using sickit-learn, in pairs.

Evaluation methods: 2h written test, can be retaken.

Evaluated skills:
- Research and Development
- Development

CM:

1. Datasets and learning (1.5 h)
2. Frequentist, Bayesian, evaluation (1.5 h)
3. Risks (1.5 h)
4. C-SVC, Lagrange formulation (1.5 h)
5. Kernels, numerical resolution (1.5 h)
6. SVMs for regression, unsupervized learning, nu-versions of SVMs. (1.5 h)
7. Arbres de décision (1.5 h)
8. Bagging (1.5 h)
9. Boosting (1.5 h)

TP:

1. Data Science en Python (3.0 h)
2. Arbres de décision (3.0 h)
3. Bagging (3.0 h)
4. Forêts aléatoires (3.0 h)
5. TP+ 1/2 (4.0 h)
6. TP+ 2/2 (4.0 h)

Deep learning

Course supervisor: Jérémy Fix

Total: 28.5 h **CM:** 13.5 h, **TP:** 15.0 h

3MD4040 *back*

Description: Deep learning is a technology that is booming, thanks in particular to the use of GPUs (Graphical Processing Units), the availability of large amounts of data and the understanding of theoretical elements that make it possible to better define neural network architectures that are more easily trainable. In this course, students will be introduced to the basics of neural networks and also to the different architectural elements that make it possible to design a neural network according to the prediction problem considered. The course is divided into modules in which questions of optimization algorithms, their initialization, regularization techniques, fully connected architectures, convolutional networks, recurrent networks, introspection techniques are addressed. Practical works on GPUs are associated with the courses.

Content: The lectures will be discussing :

- Historical introduction to neural networks, linear classifier/regressor (1.5 HPE) - Computational graph and gradient descent, Fully connected networks, RBFs, Auto-encoders, Optimisation methods, initialisation, regularisation (3 HPE) - Convolutional networks: architectures (1.5 HPE) - Convolutional networks: classification, object detection, semantic segmentation (1.5 HPE) - Recurrent networks: architectures and training (1.5 HPE) - Recurrent networks: applications (1.5 HPE) - Transformer approaches (1.5 HPE) - Generative models: autoregressive models, VAE, GANs and diffusion (1.5 HPE)

The praticals will be on:

- Introduction to pytorch on classification with linear predictors, fully connected networkjs and convolutional networks (3 HPE) - Convolutional neural networks for semantic segmentation (3 HPE) - Recurrent neural networks applied to sequence to sequence translation (3 HPE) - Adversial neural networks (3 HPE) - Self-supervized learning (3 HPE)

Learning outcomes: Being able to implement and deploy a deep learning algorithm Being able to choose the right architecture that suits a particular machine learning problem Being able to diagnose the training of a neural network (what is it learning ? how is it learning ? is it learning ? will it be able to generalize ?)

Teaching methods: The course is structured into lectures during which we introduce the theoretical and experimental notions are introduced and illustrated with various examples. Practicals allow the students to put into practice the notions we discuss during the lectures.

Means: We will be using the Pytorch framework. The students could work in pairs and will make use of GPUs of the Data Center d"Enseignement of the Metz campus for running their codes. A page will be dedicated on edunao. Forums will be opened, allowing the students to ask questions on the lectures or the tutorials, having the possibility to interact with the teaching staff but between them as well.

Evaluation methods: The assessment will be based on two elements: assessments on paper questionnaires at the beginning of each practical session and participation to a dedicated challenge in a team. The grade depends both on their submission, and a recorded video explaining their approach and results.

Evaluated skills:

- Modelling
- Research and Development
- Management

External resources:

- Site du cours
- Sujets des TPs

CM:

1. Introduction et réseaux linéaires (1.5 h)
2. Graphe de calcul, descente de gradients et réseaux feedforward (1.5 h)
3. Optimisation, Initialisation, Régularisation (1.5 h)
4. Eléments d'architectures des réseaux convolutifs (1.5 h)
5. Applications des réseaux convolutifs à la détection d'objets et la segmentation sémantique (1.5 h)
6. Les approches par transformers (1.5 h)
7. Eléments d'architecture des réseaux récurrents (1.5 h)
8. Architectures et applications des réseaux récurrents (1.5 h)
9. Modèles génératifs : modèles auto-regressifs, VAE, GANs et diffusion (1.5 h)

TP:

1. Introduction à PyTorch par la classification d'images (3.0 h)
2. Réseaux convolutifs pour la segmentation sémantique (3.0 h)
3. Application des RNNs pour la retranscription de la parole (3.0 h)
4. Modèles génératifs (3.0 h)
5. Apprentissage auto-supervisé (3.0 h)

HPC on a single computing node

Course supervisor: Stéphane Vialle

Total: 22.5 h **CM:** 9.0 h, **TD:** 3.0 h, **TP:** 9.0 h

SPM-INF-012 *back*

Description: Parallel computing on a multi-core computer using multithreading, vectorization, and serial optimization. This course presents the generic architecture of modern CPUs (multi-core, SIMD units, cache hierarchy), then efficient memory access management to effectively exploit caches, then loop vectorization to take advantage of SIMD units, and finally multithreading with OpenMP to exploit all the cores of the machine. Aspects of parallel programming and algorithms form the core of the course, accompanied by concepts of performance measurement and analysis.

Evaluation methods: 1h30 written test (can be retaken) and labwork reports (cannot be retaken).

Evaluated skills:
- Development
- Modelling

CM:

1. Cours d'architecture des CPU multi-coeurs et des caches (1.5 h)
2. Cours de compilation optimisée et de vectorisation (1.5 h)
3. Cours de mesure et d'analyse de performances (1.5 h)
4. Cours d'OpenMP et de bibliothèques BLAS (3.0 h)
5. Cours d'algorithmique parallèle (1.5 h)

TD:

1. TD d'optimisation sérielle et de vectorisation (1.5 h)
2. TD de programmation OpenMP (1.5 h)

TP:

1. TP d'optimisation sérielle et de vectorisation (3.0 h)
2. TP de programmation OpenMP (3.0 h)
3. TP d'algorithmique et de programmation parallèle optimisée (3.0 h)

Big Data computation models

Course supervisor: Stéphane Vialle

Total: 21.0 h **CM:** 10.5 h, **TD:** 1.5 h, **TP:** 9.0 h

3MD4130 *back*

Description: The goal of this course is to teach students how to develop high-performance data analysis applications in the Spark environment on distributed platforms (clusters and clouds). Distributed file system mechanisms such as HDFS will be studied, as well as Spark's extended map-reduce programming model and algorithmic on top of Spark "RDD", followed by higher-level programming models on top of Spark "Data Frames", and finally programming models on Clouds. Scaling criteria and metrics will also be studied. Throughout the course, implementations will take place on clusters and in a Cloud, and the developed solutions will be evaluated by the performance obtained on test cases, and by their ability to scale.

Content: Emergence of Big Data technologies: motivations, industrial needs, main players. Hadoop software stack, architecture and operation of its distributed file system (HDFS) Spark distributed computing architecture and deployment mechanism Spark "RDD" programming model and algorithmics of Spark extended map-reduce Spark "Data Frames" programming model applied to graph analysis (GraphX module) Architecture et environnement d'analyse de données sur Cloud Experiments and performance measures Performance criteria and metrics

Learning outcomes: After this course, students will be able:

Learning Outcome AA1: to design and implement extended map-reduce algorithms, powerful and scaling on distributed platforms, Learning Outcome AA2: to analyse the scaling capabilities of an application, Learning Outcome AA3: to use a cluster or a cloud to achieve large scale data analysis, Learning Outcome AA4: to synthetically present a data analysis solution designed on top of a "map-reduce" model.

Teaching methods: This course links 3 parts relating to "Big Data" computing models: the first on PC clusters, the second in the Cloud, and the third which assesses "scaling-up" solutions.

Course plan in 4 parts:

Part 1: Software architecture and development with Spark RDD on top of HDFS and PC clusters.

Part 2: Criteria and metrics for performance and scaling.

Part 3: Large scale computation and data analysis on Cloud.

Part 4: Development with Spark Data Frames on top of HDFS and PC clusters.

Means: Teaching team: Stéphane Vialle and Gianluca Quercini (CentraleSupelec), Wilfried Kirschemann (ANEO) Development and execution plateform: computing clusters of the Data Center for Education (DCE) of CentraleSupelec Metz campus access to a professional cloud Development environment: Spark+HDFS on DCE machines other environment on Cloud ressources

Evaluation methods: Evaluation from Labs:

The reports of the Labs will be evaluated (the content and the number of pages of the reports will be constrained, in order to force the students to an effort of synthesis and clarity) In case of unjustified absence from a practical course, a mark of 0 will be applied; in case of justified absence, the practical course will not be included in the final mark. The remedial exam will be a 1 hour written exam, which will constitute 100

Evaluated skills:

- Development
- System

CM:

1. Emergence du Big Data et technologie HDFS d'Hadoop (1.5 h)
2. Technologie Spark-RDD et programmation Map-Reduce (1.5 h)
3. Optimisation et déploiement de codes Map-Reduce (1.5 h)
4. Métriques de passage à l'échelle et architecture des Data Lakes (1.5 h)
5. Spark Data-Frames et Spark SQL (1.5 h)
6. Problématiques du cloud (1.5 h)
7. Environnement de développement sur cloud (1.5 h)

TD:

1. Algorithmique Map-Reduce en Spark (1.5 h)

TP:

1. Map-Reduce sur cluster Spark-HDFS: programmation et performances (3.0 h)
2. Analyse de données en Spark Data-Frame et Spark SQL sur cluster Spark-HDFS (3.0 h)
3. Développement et déploiement d'une analyse de données extensible sur Cloud (3.0 h)

Software engineering

Course supervisor: Michel Ianotto

Total: 36.0 h **CM:** 12.0 h, **TD:** 3.0 h, **TP:** 18.0 h

SPM-INF-011 *back*

Description: This course covers the methods and tools used to support the different phases of the software development lifecycle. It is divided into three parts: (1) Software modeling and architecture: UML and architectural design patterns; (2) Software quality: criteria, various types of testing and measurement, and testing strategies; and (3) Software project management: software lifecycles, agile methodologies, and DevOps.

Learning outcomes: At the end of this course, students will have learned the main concepts related to software development methods (agile methodologies, design patterns, software testing, etc.).

Evaluation methods: 3h written test, can be retaken

Evaluated skills:
- Modelling
- Development
- System

CM:

1. modélisation et diagrammes UML (3.0 h)
2. patrons de conception (3.0 h)
3. qualité logicielle (3.0 h)
4. projet logiciel (1.5 h)
5. méthodes agiles (1.5 h)

TD:

1. modélisation (1.5 h)
2. système à refactorer (1.5 h)

TP:

1. UML (3.0 h)
2. refactoring par application de patrons (3.0 h)
3. tests unitaires et plus (4.0 h)
4. intégration continue (4.0 h)
5. scrum (4.0 h)

Langage processing

Course supervisor: Benoît Valiron

Total: 21.0 h **CM:** 7.5 h, **TD:** 4.5 h, **TP:** 9.0 h

SPM-INF-014 *back*

Description: The objective is to introduce the concepts of language theory and put them into practice so that students fully understand and see the value of these concepts in dealing with concrete problems. We will cover regular languages, grammars, and how a compiler works.

Learning outcomes: By the end of this course, students will have understood the relevance of language theory in computer science.

Evaluation methods: Assessment based on participation in experiments (TD/TP) and results obtained

Evaluated skills:
- Certification

CM:

1. Expressions régulières, langage reconnu par un automate (3.0 h)
2. Grammaires (1.5 h)
3. Descente récursive, parseur LL (1.5 h)
4. Fonctionnement d'un compilateur (1.5 h)

TD:

1. Grammaires (1.5 h)
2. Descente récursive, parseur LL (1.5 h)
3. Compilation (1.5 h)

TP:

1. Reconnaissance de tokens dans un document (3.0 h)
2. Evaluation d'expressions arithmétiques (3.0 h)
3. Mise en oeuvre du compilateur sur machine (3.0 h)

Operations research

Course supervisor: Nicolas Jozefowiez

Total: 19.5 h **CM:** 6.0 h, **TP:** 12.0 h

SPM-INF-013 *back*

Description: The course is an introduction to combinatorial optimization problems in the applied context of operations research. The latter covers formal optimization methods useful to organizations (companies, etc.) for making rational decisions, whether in logistics, strategy (investment plans, etc.), economics, etc. This course and the optimization course are inseparable in the sense that they divide up the presentation of the various optimization techniques between them, depending on the nature of the problems these techniques solve: while the optimization course focuses on numerical analysis methods that optimize a continuous function defined in a Euclidean space, RECHOP addresses combinatorial optimization problems in a discrete space, the solution of which is more algorithmic. As a result, this course places a strong emphasis on algorithms and their complexity and can therefore be seen as an extension of the first-year course "Data Structures and Algorithms" (DSA). The course will begin by addressing combinatorial problems within the very general framework of constraint programming. We will then focus on instances of problems for which a polynomial-time solution exists, such as dynamic programming or graph matching problems. We will also develop efficient approximation algorithms for the difficult problem of integer linear programming after linear programming has been covered in the optimization course, thus linking the two courses.

Learning outcomes: At the end of this course, students will have a general understanding of the types of problems encountered in operational research. They will be able to formalize an operational research requirement and recognize the precise nature of the underlying combinatorial problem. Where appropriate, they will be able to match this combinatorial problem with a suitable solution technique.

Evaluation methods: 1h30 written test, can be retaken

Evaluated skills:
- Modelling
- Research and Development

CM:

1. Programmation par contraintes (1.5 h)
2. Problèmes d'optimisation temporelle. Programmation dynamique (1.5 h)
3. Problème d'affectations: couplage de graphes et mariages stables (1.5 h)
4. Programmation linéaire entière (1.5 h)

TP:

1. Programmation par contraintes (3.0 h)
2. Programmation dynamique (3.0 h)
3. Problèmes de couplage (3.0 h)
4. Programmation linéaire entière (3.0 h)

Optimization

Course supervisor: Michel Barret

Total: 25.0 h **CM:** 10.5 h, **TD:** 4.5 h, **TP:** 9.0 h

SPM-MAT-004 *back*

Description: In this course, students are expected to acquire and master various fundamental aspects of continuous optimization. The following concepts will be covered and implemented in practice: formulation of optimization problems, conditions for the existence of global and local minimizers, convexity, duality, Lagrange multipliers, first-order methods, linear programming. The use of differentiable programming will be presented in practical work. Stochastic gradient-free methods, such as CMAES and PSO, will also be covered.

Learning outcomes: By the end of this course, students will master the fundamental concepts of continuous optimization (conditions for the existence of global and local minimizers, convexity, duality, Lagrange multipliers, first-order methods, linear programming, stochastic methods).

Evaluation methods: 1h written test, can be retaken

Evaluated skills:
- Modelling
- Research and Development

CM:

1. Bases de l'optimisation 1/2 (1.5 h)
2. Bases de l'optimisation 2/2 (1.5 h)
3. Convexité, quelques algorithmes itératifs (1.5 h)
4. Dualité (1.5 h)
5. Programmation linéaire (1.5 h)
6. Méthode des multiplicateurs de Lagrange (1.5 h)
7. Méthodes stochastiques gradient-free (1.5 h)

TD:

1. Dualité (1.5 h)
2. Programmation linéaire (1.5 h)
3. Méthode des multiplicateurs de Lagrange (1.5 h)

TP:

1. Bases de l'optimisation (3.0 h)
2. Convexité, quelques algorithmes itératifs (3.0 h)
3. Méthodes stochastiques gradient-free (3.0 h)

Information Theory

Course supervisor: Paul Fraux

Total: 11.0 h **CM:** 4.5 h, **TD:** 1.5 h, **TP:** 3.0 h

SPM-MAT-007 *back*

Description: In this course, students are expected to acquire and master the basics of Information theory. The following concepts will be covered : source coding problematics, statistical sample and probability distributions entropy, joint and conditional entropy, mutual information and Kullback-Leibler divergence and entropy maximum principle. The use of prefix trees to design uniquely decodable codes will be illustrated through the example of Huffman's optimal coding algorithm.

Learning outcomes: At the end of this course, students will be able to describe and explain the fundamental concepts of source coding and the associated issues; provide an intuitive understanding of the implications of the course concepts for the source coding problem; manipulate and analyze entropy from a mathematical perspective; define the Kullback–Leibler divergence and relate it to the notions of entropy and mutual information; and apply the principle of maximum entropy.

Evaluation methods: 2h writtent test, can be retaken

CM:

1. Codage source de canaux discrets (1.5 h)
2. Théorie de l'information 1/2 (1.5 h)
3. Théorie de l'information 2/2 (1.5 h)

TD:

1. Entropie et divergence KL (1.5 h)

TP:

1. Codage de Huffman (3.0 h)

Business Management and Transformation

Course supervisor: Francis Dorveaux

Total: 26.5 h **CM:** 5.5 h, **TD:** 12.0 h, **TP:** 8.5 h

SPM-HEP-014 *back*

Description: The objective of this course is to explore the diversity of businesses and their integration into a global socio-economic and environmental context, as well as the dynamics they generate. The pedagogical approach combines theoretical input through lectures with a group fieldwork investigation, allowing future entrepreneurs to be confronted with the realities of business creation and development.

Learning outcomes: By the end of this course, students will have gained a comprehensive understanding of the entrepreneurial ecosystem and the forces that drive it.

Evaluation methods: MCQ (30 min), individual oral interactions

Evaluated skills:
- Business Intelligence

CM:

1. Introduction (1.0 h)
2. Panorama des entreprises (2.0 h)
3. Synthèse et débat (2.5 h)

TD:

1. TD 1 : Les théories modernes de l'entreprise + explications des attendus des TD 2 à 6 (1.5 h)
2. TD 2 : Les processus "colonne vertébrale" de l'organisation (1.5 h)
3. TD 3 : La dynamique externe (1.5 h)
4. TD 4 : Transformation des entreprises (3.0 h)
5. TD 5 : Business Development 1 (à préciser) (1.5 h)
6. TD 6 : Business Development 2 (à préciser) (3.0 h)

TP:

1. 1ère visite en entreprise (4.5 h)
2. 2ème visite en entreprise (4.0 h)

Conferences

Course supervisor: Damien Rontani, Hervé Frezza-Buet

Total: 10.0 h **CM:** 10.0 h

SPM-HEP-013 *back*

Description: This course consists of a series of seminars.

Learning outcomes: By the end of these lectures, students will have broadened their perspective on environmental, economic, societal, and ethical issues, depending on the expertise of the speakers.

CM:

1. tdb (10.0 h)

Human ressources Management

Course supervisor: Damien Rontani, Hervé Frezza-Buet

Total: 18.0 h **TD:** 18.0 h

SPM-HEP-019 *back*

Description: This course has a dual objective: on the one hand, to equip students with knowledge of their rights and obligations in relation to HR practices, particularly during recruitment; and on the other hand, to enable engineers transitioning into managerial roles to collaborate effectively with management and HR departments.

Learning outcomes: By the end of this course, students will have acquired a foundational understanding of labor law and HR functions, which will be useful both as employees and in their supervisory or managerial responsibilities.

Evaluated skills:
- Management

TD:

1. tbd (18.0 h)

Sport S07

Course supervisor: Hervé Frezza-Buet

Total: 21.0 h **TD:** 21.0 h

SportS07 *back*

TD:

1. Cours de sport (21.0 h)

Foreign Languages and Culture 1

Course supervisor: Elisabeth Leuba

Total: 21.0 h **TD:** 21.0 h

LV1S07 *back*

Description: The first foreign language is generally English. Students are divided into level groups ; in class, work is not only focused on the 4 language competences but also on various topics studied in depth according to students' levels. Topics cover a range of fields, such as civilisation, society and the professional world. Limited class size enables active participation and significant improvement in the language. The educational approach is varied: group work, class presentations, specific exercises, research, debates, etc.

Learning outcomes: At the end of the course, students will have improved their ability to communicate in an international professional, academic or personal context.

Evaluation methods: Assessment will be by continuous assessment according to criteria to be determined by each teacher, taking into account personal investment in the course. Each course will be marked out of 20 at the end of the semester.

Evaluated skills:
- Research and Development
- Consulting
- Business Intelligence
- Management

TD:

1. Cours (21.0 h)

Foreign Languages and Culture 2

Course supervisor: Beate Mansanti

Total: 21.0 h **TD:** 21.0 h

LV2S07 *back*

Description: Students are offered a range of second foreign languages at different levels, including for beginners. Students are divided into level groups; in class, work is not only focused on the 4 language competences but also on various topics studied in depth according to students' levels. Topics cover a range of fields, such as civilisation, society and the professional world. Limited class size enables active participation and significant improvement in the language. The educational approach is varied: group work, class presentations, specific exercises, research, debates, etc.

Learning outcomes: At the end of the course, students will have improved their ability to communicate in an international professional, academic or personal context.

Evaluation methods: Assessment will be by continuous assessment according to criteria to be determined by each teacher, taking into account personal investment in the course. Each course will be marked out of 20 at the end of the semester.

Evaluated skills:
- Research and Development
- Consulting
- Business Intelligence
- Management

TD:

1. Cours (21.0 h)

Application Architectures

Course supervisor: Virginie Galtier

Total: 41.0 h **CM:** 12.0 h, **TD:** 1.5 h, **TP:** 25.5 h

SPM-INF-015 *back*

Description: This course provides an overview of application architectures (historical background, principles, benefits and limitations, usage contexts, etc.). Two architectures will be examined in greater depth: REST architecture (with OpenAPI) and message-oriented middleware (MOM)–based architectures (Kafka). Students will be introduced to fault tolerance and scalability issues, and will discover a deployment solution (Kubernetes).

Learning outcomes: By the end of this course, students will be able to analyze an architectural proposal and implement and deploy two standard architectures in a simple context.

Evaluation methods: 2h written test, can be retaken

Evaluated skills:
- Development
- System

CM:

1. panorama des architectures applicatives (3.0 h)
2. architecture REST, OpenAPI (1.5 h)
3. architecture basée sur les MOM (Kafka) (1.5 h)
4. tolérance aux pannes, passage à l'échelle et déploiement (K8s) (3.0 h)
5. cloud (3.0 h)

TD:

1. études de cas architecture (1.5 h)

TP:

1. client REST (3.0 h)
2. serveur REST (3.0 h)
3. tests avec SoapUI (3.0 h)
4. graphQL (1.5 h)
5. tuto Kafka (4.5 h)
6. Kubernetes (4.5 h)
7. cloud 1/2 (3.0 h)
8. cloud 2/2 (3.0 h)

Software development project

Course supervisor: Hervé Frezza-Buet

Total: 84.0 h **Projet:** 84.0 h

SPM-PRJ-003 *back*

Description: This module focuses on applying software engineering methods to develop a large-scale IT project. This project may involve industrial collaborations and joint work with other specialties. The aim of the module is to gain experience in designing and implementing a project in a situation that is as close as possible to a professional environment.

Learning outcomes: At the end of this course, students will have implemented a professional methodology for carrying out a significant IT development project as part of a team.

Evaluation methods: Report and defense

Evaluated skills:
- Modelling
- Research and Development
- Development
- Management

Introduction to Quantum Computing

Course supervisor: Damien Rontani, Stéphane Vialle

Total: 35.5 h **CM:** 16.5 h, **TD:** 4.5 h, **TP:** 9.0 h, **Projet:** 5.5 h

SPM-INF-004 *back*

Description: This Quantum Programming and Algorithms course offers a comprehensive dive into the fundamentals and practical applications of quantum accelerators. Students will explore contemporary quantum architectures, including the principles of analog and digital architectures, as well as innovations such as hybrid CPU-QPU, NISQ, and LSQ architectures. The course will cover the formalism of qubits and digital quantum programming, highlighting the importance of superposition and entanglement for quantum computations. The principles and methods of measuring results will also be discussed. An introduction to quantum circuits, including basic gates and early circuits, will enable students to understand the practical basics of quantum programming. Using tools such as QFT, Grover, QPE, and QMC, students will explore classical circuits and their applications, while examining the limitations on NISQ architectures. Finally, students will delve into variational circuits and algorithms, including QAOA and Vxx circuits, and study runtime and performance models for QPUs as well as CPU-QPU loops, familiarizing themselves with the orders of magnitude of current runtimes.

Bibliography:
- Ref. [1] : R. Hundt, Quantum Computing for Programmers, Cambridge University Press (2022)
- Ref. [2] : P. Kaye, R. Laflamme, M. Mosca, An Introduction to Quantum Computing, Oxford University Press (2006)

Learning outcomes: AA1: Identify basic gates and build initial quantum circuits – AA2: Effectively use tools such as QFT, Grover, QPE, and QMC algorithms to solve engineering problems – AA3: Understand and be able to assess the limitations of NISQ architectures and propose appropriate solutions – AA4: Design and implement variational circuits and algorithms, such as QAOA and Vxx – AA5: Evaluate the performance and execution times of quantum algorithms in various contexts, including QPUs and CPU-QPU loops.

Evaluation methods: Assessment of the mini project

Evaluated skills:
- Modelling
- Development

CM:

1. Cours d'architectures quantiques (1.5 h)
2. Cours de formalisme pour l'informatique quantique digitale (3.0 h)
3. Cours d'introduction aux portes et circuits quantiques (3.0 h)
4. Cours de présentation des circuits quantiques classiques (4.5 h)
5. Cours de modèles de temps d'exécution et de performance (1.5 h)
6. Cours de présentation des circuits quantiques variationnels (3.0 h)

TD:

1. TD de formalisme et d'analyse de circuits quantiques (1.5 h)
2. TD de conception d'algorithmes quantiques sur QPU (1.5 h)
3. TD de conception d'algorithmes variationnels sur CPU+QPU (1.5 h)

TP:

1. TP de mise en oeuvre de circuits quantiques en qiskit sur simulateur et machines quantiques (3.0 h)

2. TP de conception et mise en oeuvre d'algorithmes quantiques à partir de circuits connus (3.0 h)
3. TP de conception d'une méthode d'optimisation par algorithme variationnel sur CPU+QPU (3.0 h)

Programming paradigms

Course supervisor: Hervé Frezza-Buet

Total: 28.5 h **CM:** 6.0 h, **TP:** 22.5 h

SPM-INF-016 *back*

Description: This course covers different programming paradigms by introducing different languages that are strongly influenced by these paradigms. We will draw on functional languages such as Haskell and Ocaml for functional approaches (the concepts of type inference and lazy evaluation), Prolog for logic programming, and LISP for construction through the calculation of executable expressions...

Learning outcomes: At the end of this course, students will have learned to think about programming in ways other than traditional methods. They will be able to recognize these paradigms when they arise implicitly and apply appropriate solutions (for example, recognizing a functional approach learned in Caml in recursive template designs in C++, recognizing a logical programming paradigm in makefiles, etc.). The idea is also to be able to propose elegant solutions to certain problems without being dependent on the structure of the language used (e.g., implementing a lazy approach in C++).

Evaluation methods: Assessment based on participation in experiments (TP) and results returned

---

CM:

1. Cours-1 (1.5 h)
2. Cours-2 (1.5 h)
3. Cours-3 (1.5 h)
4. Cours-4 (1.5 h)

TP:

1. TP-1 (4.5 h)
2. TP-2 (4.5 h)
3. TP-3 (4.5 h)
4. TP-4 (4.5 h)
5. TP-5 (4.5 h)

Autonomous robotics

Course supervisor: Francis Colas

Total: 41.0 h **CM:** 13.5 h, **TD:** 3.0 h, **TP:** 22.5 h

SPM-INF-017 *back*

Description: This course will introduce the field of autonomous robotics (vehicle driving, exploration and inspection robots, etc.) by showing how this issue integrates a wide range of technologies (localization (SLAM), point clouds, planning, pattern recognition) and how this integration is achieved at the system level (illustrations with ROS). The laboratory work associated with the course will be carried out on the robots of the Metz campus robotics platform and in simulation. This work will provide an opportunity to integrate different machine learning and signal processing techniques on robots moving in their environment. The course and practical applications will allow students to discover these techniques in real-world cases.

Prerequisites: It is necessary to have a good knowledge of Python programming and a sound knowledge of probability. The practical work also requires a minimum level of familiarity with Linux.

Learning outcomes: At the end of this course, students will be familiar with ROS and the key concepts of autonomous robotics. They will have experienced the difficulty of coupling information processing with a system operating in a real environment and will have gained experience in implementing solutions, guided by the methodology induced by ROS.

Teaching methods: The lectures introduce the essential concepts and algorithms for autonomous robotics (13.5 HPE), supplemented by tutorials (3 HPE) and extensively supported by practical work in simulation and on real platforms (22.5 HPE).

Means: Tutorials, consisting of exercises, will enable students to apply the concepts covered in class. Practical work will enable students to programme and test the algorithms covered in class in simulations and on real robotic platforms using the ROS2 framework.

Evaluation methods: 2h written exam, can be retaken

CM:

1. Intro (1.5 h)
2. Introduction à ROS (1.5 h)
3. Rappels de probabilités (1.5 h)
4. Estimation d'état (1.5 h)
5. Localisation (1.5 h)
6. Carto + SLAM (1.5 h)
7. Planif (1.5 h)
8. Navigation (1.5 h)
9. Architecture et interaction (1.5 h)

TD:

1. Filtres de Kalman (1.5 h)
2. Localisation (1.5 h)

TP:

1. ROS et simulation (3.0 h)
2. Navigation et robots réels (3.0 h)
3. Filtre de Kalman et estimation d'état (1.5 h)
4. Localisation (3.0 h)

5. Carto + SLAM (3.0 h)
6. Path planning (3.0 h)
7. Path following (3.0 h)
8. Integration (3.0 h)

Robotic and AI project

Course supervisor: Hervé Frezza-Buet

Total: 21.0 h **Projet:** 21.0 h

SPM-PRJ-004 *back*

Description: The objective of this project is to integrate artificial intelligence algorithms into robotic platforms in order to confront the reality of situated artificial intelligence approaches. It will involve synthesizing the skills acquired in the seventh semester courses and in the robotics course. In addition, emphasis will be placed on experimental rigor (implementation, reporting) in order to prepare students to report on research experiments.

Learning outcomes: At the end of this course, students will have practiced AI outside of manipulating datasets, but on dynamic systems in interaction.

Evaluation methods: Evaluation based on the code deposited on the git and ongoing monitoring by the supervisors

Evaluated skills:
- Modelling
- Research and Development
- Development
- Management

Engineer, environment and society

Course supervisor: Julien Colin

Total: 14.0 h **TD:** 14.0 h

SPM-HEP-011 *back*

TD:

1. tbd (14.0 h)

Controversy

Course supervisor: Hervé Frezza-Buet, Damien Rontani

Total: 18.0 h **TD:** 18.0 h

SPM-HEP-018 *back*

Description: The objective of this course is to contribute to students' civic and critical education, enabling them to develop the ability to analyze public debates and engage constructively with differing opinions.

Learning outcomes: By the end of this course, students will be able to recognize the dimensions of a controversy and will be familiar with several techniques for working towards its resolution.

TD:

1. tbd (18.0 h)

Systems engineering

Course supervisor: Virginie Galtier

Total: 21.5 h **CM:** 11.0 h, **TP:** 9.0 h

SPM-HEP-017 *back*

Description: This course introduces the engineering of complex systems, with an emphasis on modern modeling and simulation approaches. It uses the MBSE (Model-Based Systems Engineering) methodology to guide system design throughout the lifecycle. Students gain hands-on experience with SysML to formalize some aspects of the requirements, functions, and structures. Complementary paradigms are also explored, including multi-agent modeling (NetLogo) for emergent behaviors and cyber-physical system modeling (OpenModelica) for simulating physical components and their interactions with control systems. An introduction to the FMI standard covers co-simulation of heterogeneous models. Finally, a presentation from AFIS and a professional testimonial illustrate the role of systems engineers in industry and the practical use of digital twins.

Learning outcomes: By the end of this course, students will be able to formulate the relevant questions for system design, particularly regarding modeling aspects, and outline possible solutions.

Evaluation methods: Multiple-choice questions and case study (1.5 hours)

Evaluated skills:
- Modelling

CM:

1. Introduction à l'ingénierie système (définitions, historique, motivations, concepts système) (2.0 h)
2. MBSE (définitions, évolution, cycles de vie, interfaces, processus, exigences, V&V (2.0 h)
3. Introduction à SysML (1.0 h)
4. modélisation et simulation à base d'agents (1.0 h)
5. modélisation de systèmes cyber-physiques et co-simulation (1.0 h)
6. AFIS (2.0 h)
7. Jumeaux Numérique (2.0 h)

TP:

1. familiarisation avec un logiciel de modélisation SysML, mise en évidence des liens entre diagrammes (3.0 h)
2. modélisation et simulation à base d'agents (3.0 h)
3. modélisation de systèmes cyber-physiques et co-simulation (3.0 h)

Sport S08

Course supervisor: Hervé Frezza-Buet

Total: 21.0 h **TD:** 21.0 h

SportS08 *back*

TD:

1. Cours de sport (21.0 h)

Engineering internship

Course supervisor: Hervé Frezza-Buet, Damien Rontani

SPM-STA-002 *back*

Evaluated skills:

- Modelling
- Research and Development
- Development
- Certification
- System
- Consulting
- Business Intelligence
- Management

Foreign Languages and Culture 1

Course supervisor: Elisabeth Leuba

Total: 21.0 h **TD:** 21.0 h

LV1S08 *back*

Description: The first foreign language is generally English. Students are divided into level groups ; in class, work is not only focused on the 4 language competences but also on various topics studied in depth according to students' levels. Topics cover a range of fields, such as civilisation, society and the professional world. Limited class size enables active participation and significant improvement in the language. The educational approach is varied: group work, class presentations, specific exercises, research, debates, etc.

Learning outcomes: At the end of the course, students will have improved their ability to communicate in an international professional, academic or personal context.

Evaluation methods: Assessment will be by continuous assessment according to criteria to be determined by each teacher, taking into account personal investment in the course. Each course will be marked out of 20 at the end of the semester.

Evaluated skills:
- Research and Development
- Consulting
- Business Intelligence
- Management

TD:

1. Cours (21.0 h)

Foreign Languages and Culture 2

Course supervisor: Beate Mansanti

Total: 21.0 h **TD:** 21.0 h

LV2S08 *back*

Description: Students are offered a range of second foreign languages at different levels, including for beginners. Students are divided into level groups; in class, work is not only focused on the 4 language competences but also on various topics studied in depth according to students' levels. Topics cover a range of fields, such as civilisation, society and the professional world. Limited class size enables active participation and significant improvement in the language. The educational approach is varied: group work, class presentations, specific exercises, research, debates, etc.

Learning outcomes: At the end of the course, students will have improved their ability to communicate in an international professional, academic or personal context.

Evaluation methods: Assessment will be by continuous assessment according to criteria to be determined by each teacher, taking into account personal investment in the course. Each course will be marked out of 20 at the end of the semester.

Evaluated skills:
- Research and Development
- Consulting
- Business Intelligence
- Management

TD:

1. Cours (21.0 h)