ISEN MÉDITERRANÉE | L'ÉCOLE DES INGÉNIEURS DU NUMÉRIQUE

ISEN YNCREA MEDITERRANEE

Course Descriptions

Language of Instruction: English

(course details are subject to change)

MODULE: ARTIFICIAL INTELLIGENCE (common to all specializations)

Course Name: Generative Artificial Intelligence

PREREQUISITE: Basic programming knowledge (Python) Basic knowledge of machine learning fundamentals (neural networks, training, etc.) Familiarity with natural language processing concepts

LEARNING OUTCOMES: Understand the foundations of generative AI, including LLMs Understand the transformer architecture and how it enables LLMs to generate human-like language Understand the training process for LLMs Understand how to fine-tune LLMs Understand the challenges and limitations of generative AI

COURSE CONTENT: This course is designed to provide an in-depth understanding of generative artificial intelligence with a focus on Large Language Models (LLMs) and Diffusion models. The course covers the foundations of LLMs, including their architecture, training, and finetuning and explores their use in natural language processing tasks such as text generation, summarization, and translation. Topics Covered: Introduction to Generative AI Foundations of LLMs Transformer Architecture Training LLMs Fine-tuning LLMs for specific tasks Natural language processing tasks with LLMs (text generation, summarization, translation, etc.) Applications of LLMs Challenges and limitations of generative AI

ASSESSMENT METHOD: Multiple choices questions test

Course Name: Dev Platform (new course 2026/2027 – syllabus pending)

PREREQUISITE: *pending*

LEARNING OUTCOMES: *pending*

COURSE CONTENT: *pending*

ASSESSMENT METHOD: *pending*

Course Name: Tools AI Level 1

PREREQUISITE: Basic C language, Python, Basic knowledge of machine learning, deep learning and generative AI

LEARNING OUTCOMES: We will explore machine learning techniques focused on algorithmic approaches, such as genetic algorithms, reinforcement learning, and optimized tree traversal, within the context of practical applications. This course aims to deepen the use of non-traditional AI approaches, such as evolutionary algorithms, reinforcement learning, and optimized decision tree traversal. The focus is on algorithmic understanding, practical implementation, and real-world applications. By the end of this course, the student will be able to design and implement genetic algorithms to solve complex problems, to develop intelligent agents using reinforcement learning, to apply and optimize decision tree traversal for intelligent decision-making, to integrate these tools into a complete AI pipeline, adapted to the application context.

COURSE CONTENT: Genetic Algorithms and Artificial Evolution, Reinforcement Learning (RL), Optimized Decision Tree Traversal

ASSESSMENT METHOD: Project

MODULE: SYSTEM DEVELOPMENT

Course Name: Systems Architecture

PREREQUISITE: General comprehension of the LINUX operating system, Basic knowledge of network architecture and information systems, Fundamental virtualization concepts recommended

LEARNING OUTCOMES: By the end of the course, the student is able to deploy and manage a Kubernetes cluster by implementing application deployment solutions using tools such as Helm, Kustomize, and GitOps, is able to integrate virtual machines into a Kubernetes environment using KubeVirt to unify the management of heterogeneous infrastructures (VMs and containers) within a single cluster.

COURSE CONTENT: This three-day Systems Architecture course immerses participants in the evolution of IT infrastructures, from physical servers to cloud-native platforms. The program combines theoretical sessions and hands-on workshops to master virtualization, containerization, and orchestration with Kubernetes and its associated tools (Helm, Kustomize, GitOps), while exploring the integration of traditional virtualization through KubeVirt.

ASSESSMENT METHOD: Practical work assessment.

Course Name: Linux Shell

PREREQUISITE: Knowledge in at least one procedural programing language - be able to differentiate interpreted language versus compiled one.

LEARNING OUTCOMES: Know the basics of LINUX operating system, acquire knowledge of a first set of SHELL commands, know how to write SHELL scripts to automate and capitalize operations on the operating system (administration or other).

COURSE CONTENT: Main chapters: reminder to the LINUX operating system, introduction to SHELL (first commands), SHELL language (control structures, iterations, variables, etc.), SHELL scripts, SHELL tools.

ASSESSMENT METHOD: Written exam and practical work assessment.

Course Name: Language Security Programming

PREREQUISITE: Fluency with at least one programming language.

LEARNING OUTCOMES: Many, many programming languages are in active use, and many more have been created. What makes them different, especially security wisdom? Why can't anyone agree with them? This module will help you answer this question and then gives an overview of good security practices in the field of software engineering.

COURSE CONTENT: Introduction to a typology of programming languages: the major paradigms will be described, with examples. Cognitive biases and language choice, how to avoid some bad choices related to our way of thinking. An attacker model: how they approach problems, and what are the impacts on the role of the system designer. The most important pitfalls in software development: what impacts do languages, communities, and tasks have on the security of the final software. Countermeasures and secure practices: an abbreviated guide to secure development. Formal methods: a very brief overview of the most powerful secure programming techniques

ASSESSMENT METHOD: Ongoing assessment and written exam.

Course Name: Labview

PREREQUISITE: Basics in algorithms, have a computer with LabVIEW2021 installed.
LEARNING OUTCOMES: Learn basics about National instruments LabVIEW and be able to use it in full autonomy and realize small projects.
COURSE CONTENT: Environment, create your first application, debugging, using loops, creating data structures, using decision-making structures, modularity, acquisition, accessing files, using state-based design.
ASSESSMENT METHOD: Written exam.

Course Name: Labview Embedded

PREREQUISITE: Know the environment LabVIEW and communication principles of low level.
LEARNING OUTCOMES: Create an application LabVIEW FPGA over module my RIO.
COURSE CONTENT: Introduction Chap 1: Introduction to real-time systems / Chap 2: Configuring your hardware / Chap 3 : Accessing your I/O / Chap 4: Real-Time Architecture / Chap 5: Inter-Process communication / Chap 6: Communication between RT target and Host / Chap 7: Verifying your application / Chap 8: Introduction to deployment
ASSESSMENT METHOD: Practical work assessment

Course Name: System Architecture UML

PREREQUISITE: General understanding of Linux/Unix operating systems, Basic knowledge of network architecture and information systems, Fundamental virtualization concepts recommended.
LEARNING OUTCOMES: Understand the evolution of IT infrastructures, from physical servers to cloud-native architectures. Deploy and manage virtual machines and containers. Orchestrate applications at scale with Kubernetes. Work with advanced orchestration tools (Helm, Kustomize, GitOps). Integrate traditional virtualization environments into Kubernetes using KubeVirt.
COURSE CONTENT: This three-day Systems Architecture course immerses participants in the evolution of IT infrastructures, from physical servers to cloud-native platforms. The program combines theoretical sessions and hands-on workshops to master virtualization, containerization, and orchestration with Kubernetes and its associated tools (Helm, Kustomize, GitOps), while exploring the integration of traditional virtualization through KubeVirt.
ASSESSMENT METHOD: Practical works assessments

Course Name: Microservice (course presents in the module Cloud & Deployment)

PREREQUISITE Fundamentals on operating systems, networks and software architecture.
LEARNING OUTCOMES Learning fundamentals and medium knowledge on Microservices architecture.
COURSE CONTENT Evolution of application architectures over time. Characteristics of microservices architecture: Componentization via services, organized around business capabilities, Smart endpoints and dumb pipes, Decentralized governance, Decentralized data management, Infrastructure automation, Design for failure, Evolutionary design, Implementation and deployment of a microservice architecture
ASSESSMENT METHOD Written exam and project.

MODULE : CLOUD DEPLOYMENT (Cyber & Dev)

Course Name: Application Programming Interface (API)

PREREQUISITE: Basics software architecture knowledge.
LEARNING OUTCOMES: This training is intended to introduce the concept of API (Application Programming Interface) and its implementation on the web with a focus on the REST architecture style. At the end of the course, the student can understand and explain the different types of APIs and how they work, particularly the HTTP protocol and REST architecture and design, implement, and secure a REST API by following best development and security practices.
COURSE CONTENT: API overview, the different APIs types, the HTTP protocol, architecture REST, implementing a REST API, securing a REST API
ASSESSMENT METHOD: Project and practical work.

Course Name: Cloud computing

PREREQUISITE: Students are expected to have a solid foundation in software development (programming, command line usage, Git), a functional understanding of IP networking (routing, TCP/UDP, HTTP, DNS), and a grasp of basic system concepts (processes, storage) and security fundamentals (authentication, authorization). Basic knowledge of virtualization (VMs or containers) is also recommended.
LEARNING OUTCOMES: At the end of the Cloud Computing course, the student will be able to design, configure, and operate a virtualized cloud architecture, leveraging core services related to compute, storage, networking, security, monitoring, automation (Infrastructure as Code), API exposure, and the integration of artificial intelligence services — all within the context of agile, secure, scalable, and resilient software development.
The student will be able to:
Define the technical components of cloud architecture (VMs, containers, structured and unstructured storage, virtual networking, APIs, etc.)
Deploy and monitor cloud environments using automation and observability tools
Implement security mechanisms (IAM, encryption, firewalls, authentication)
Integrate ready-to-use AI services (vision, language, prediction) to enhance cloud applications. Apply principles of elasticity, high availability, and cost optimization
Operate cloud services within a professional development workflow (collaborative, reproducible, well-documented)
COURSE CONTENT: Learning what a cloud is, start a solid platform, use Google Cloud to build apps, learning what an API is.
ASSESSMENT METHOD: Ongoing assessment, written exam and practical work

MODULE : MICROCONTROLER & INTERFACES

Course Name: STM32 Level 1 +AI

PREREQUISITE: C language

LEARNING OUTCOMES: Students will discover ARM cortex processors and microcontroller-related principles through practice and lecture. They will use the Nucleo STM32L152RE demo board which embeds the STM32 MCU (Low power family). The goal is to understand code execution & hardware principles. They will configure registers through HAL library given by manufacturers to implement peripherals-related principles (Clock tree, GPIO, Timers, SPI).

ASSESSMENT METHOD: Practical work.

Course Name: Sensors

PREREQUISITE: No specific prerequisite.

LEARNING OUTCOMES: This course allows students to know the basics of an embedded system and associated data buses. The course is punctuated with various corrected exercises to better understand the concepts covered.

COURSE CONTENT: Part 1 - Embedded Systems - Basics

1) Overview; 2) Processors; 3) Architectures; 4) Tools and Peripherals; 5) Terms

Part 2 - Embedded Data Buses

1) Overview; 2) Internal buses; 3) External buses; 4) Dedicated buses

ASSESSMENT METHOD: Written exam

Course Name: STM32 Edge AI

PREREQUISITE: Knowledge and basic experience in embedded development, preferably on STM32 microcontrollers. Basic knowledge in artificial intelligence.

LEARNING OUTCOMES: Know possible use-cases of EdgeAI and their application fields. Know principles of Machine Learning and Neural Networks. Know how to train a Convolutionnal Neural Network using the "STM32 AI Model Zoo" and custom datasets to perform transfer learning. Know how to evaluate model performance on validation and test datasets by analyzing training curves and confusion matrices. Know how to quantize a model to reduce its size and accelerate its execution. Know how to estimate the model execution time on a selected MCU. Know how to deploy the model on the MCU using a provided application code. Know how to test the model on device with sounds and discuss coherence of the results. Understand the interests of using an AutoML solution. Understand and apply a classic EdgeAI project development flow.

COURSE CONTENT: The course is divided into 2 practical work sessions complemented with more theoretical aspects. In the first practical work session, students will learn what is a neural network and discover main aspects of its architecture. Then, they will follow a workflow to deploy a Convolutional Neural Network performing audio event classification. They will start with a provided model to train it, evaluate its performance, estimate its execution time, optimize it and test it on the provided STM32 board (B-U585I-IOT02A). Their goal will be to understand the role of each step and be able to interpret the results they will get. Then, they will generate an AI model by themselves and try to make it more efficient than the provided one to perform the audio event classification. In the second practical work session, students will learn some information about machine learning, why and how to use it today in microcontrollers. Then, they will solve a problem by generating a machine learning model

from data they will collect by themselves. They will test the model and deploy it on the provided STM32 board (B-U585I-IOT02A). They will have to code the AI model integration in a provided STM32CubeIDE project to make the embedded model work as expected.
ASSESSMENT METHOD: Written exam

MODULE: SOFTWARE DEVELOPMENT

Course Name: Advanced Java

PREREQUISITE: Basics in object-oriented development and more particularly in Java
LEARNING OUTCOMES: Master the concept of object-oriented programming and have a solid foundation in Java.
COURSE CONTENT: Reinforcement of the fundamentals in Java and manipulation of this language for different uses.
Revision of the concept of object-oriented programming. Using Java to create a rest API. Creation of a client server in java to imitate the BLE connection
ASSESSMENT METHOD: Assessment through exams.

Course Name: C++

PREREQUISITE: Basic programming notions, knowledge of Java or C.
LEARNING OUTCOMES: Understand major concepts of object-oriented programming. Experience in C++ programming, and program development within an integrated development environment. Learn syntax, features of, and how to utilize the Standard Template Library.
COURSE CONTENT: This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, encapsulation, inheritance, polymorphism, templates, container classes, Standard Template Library, standard C++, program organization. It also includes the comparison of C++ with Java. This course has a strong practical emphasis, and students will be required to implement Object-Oriented concepts in C++ during tutorial sessions and project. A simple object-oriented methodology is detailed to design the project using UML.
ASSESSMENT METHOD: Assessment through exams.

Course Name: Software Engineering

PREREQUISITE: Algorithm; UML
LEARNING OUTCOMES: Software engineering is a fundamental discipline for the design, development and maintenance of complex and robust software systems. This course aims to provide the knowledge and skills needed to effectively address the challenges encountered in projects.
COURSE CONTENT: Understanding of fundamental concepts: Deepen the principles of software engineering, including software life cycles (waterfall models, agile, DevOps, etc.). Analyze and specify functional and non-functional requirements.
Mastery of development methodologies: Discover and apply modern approaches to project management and software design. Integrate quality assurance methodologies, including unit testing, integration testing and code reviews.
Software design and architecture:

Explore software architecture principles, design patterns and modern paradigms (object-oriented, microservices, etc.). Learn design tools and frameworks such as UML
Practice of collaborative development: Use version control tools (Git) and team collaboration.
Advanced Perspectives: Address emerging topics such as Test-Driven Development (TDD), Model-Driven Engineering (MDE).
Introduction to Software Engineering; Software Development Process; Agile Development; Requirements Engineering; System Modeling; Architectural Design; Software Design and Implementation; Validation & Testing; Collaborative Work
ASSESSMENT METHOD: Assessment through exams and Work at home.

MODULE: IoT ENGINEERING SYSTEMS

Course Name: RTOS

PREREQUISITE: C programming language, STM32
LEARNING OUTCOMES: Understand and master RTOS type systems
COURSE CONTENT: real-time operating systems. Why real time. Critical zone and mutual exclusion. Tasks. Communication and priorities. Communication between tasks. Queue management.
ASSESSMENT METHOD: Written exam.

Course Name: Power – Consumption management

PREREQUISITE: This course requires knowledge of embedded systems and embedded c code.
LEARNING OUTCOMES: The main objective of this course is to be able to develop a device for the Internet of Things whose autonomy is several years when powered by a button cell battery (CR2032). This implies achieving an average power consumption of less than 100 µW, which requires a mastery of ultra-low power consumption modes in embedded systems.
COURSE CONTENT: Internet of Things Introduction (Architectures, Communication, Sensors, ...). STM32 is handed on for ultra-low power modes. Batteries. Project on power management.
ASSESSMENT METHOD: Project and report.

Course Name: PENDING

More information on this new course will be available soon

Course Name: Architecture UML IoT

PREREQUISITE: Fundamentals on object-oriented programming, basics on Java language.
LEARNING OUTCOMES: Understand the fundamental principles of software architecture and software design. Understand the importance of defining a software architecture. Design robust, scalable and maintainable software systems. Understand how to move from a specification to a software architecture.
COURSE CONTENT: Introduction - Definitions and key concepts. Importance of software architecture.
Definition of a domain model - UML modelling.

Architectural styles - Architectural patterns: horizontal and vertical layers, MVC, etc. Web Architecture, Micro-services Architecture (MSA).
Modelling - UML, class diagrams, package diagrams, component and deployment diagrams.
Design - SOLID principles. Introduction to GoF (Gang Of Four) design patterns.
- Resources :
  o Modelio UML modelling tools
ASSESSMENT METHOD: Practical work based on a case study and theoretical exam (2H)

MODULE: E-HEALTH

Course Name: Introduction to E-Health

PREREQUISITE: Basics of electronics, basics of computer science, interest in software and hardware technologies.
LEARNING OUTCOMES: This introduction to e-health presents the contribution of digital technology to the health and well-being of users and patients. It covers the use of digital information transmission, management and sharing tools such as electronic medical records, electronic health records and personal health records, for the benefit of both medical and medico-social practices. This introduction also covers medical devices, their fields of application and classifications, the support of microsystems and their embedded electronics to the development of new types of portable microsensors, surgical instruments, drug delivery systems or laboratories on a chip.
COURSE CONTENT: Introduction: Applications of information and communication technologies to the health and well-being of users and patients. Different types of health data: Electronic Medical Record (EMR) - Electronic Health Record (EHR) - Personal Health Record (PHR). Health Data Applications: Sensors (DMP), Diagnostics and Guidance, Teleconsultation and Telemedicine, Remote Medical Sensor Tracking, Databank and Statistical Analysis. Medical devices: definition, classes, CE certification, development cycles. Miniaturized and microsystem devices: Evolution of technologies, medical tricorder (health companion), micro devices and bio-MEMS (Micro Electromechanical Systems), basic elements - sensors and actuators -, implanted and microsurgical devices. Microfluidics and laboratories on chip: General perspectives, basic microfluidic operations (pumps, valves, mixer, etc.), devices for biomolecular characterization, devices for bio-cellular studies, organs on chips. Synthesis with examples of application for cancer research.
ASSESSMENT METHOD: Assessment through exams

Course Name: Molecular biology

PREREQUISITE: Basics in chemistry, basics in life sciences, interest in biomedical sciences and medicine.
LEARNING OUTCOMES: This course aims to bridge the gap between the fields of engineering and biomedical sciences. It is structured into three sections. The first section introduces cellular organization and its main functions. The second section explores biomolecular structures and interactions, including the structure of DNA, transcription and the role of mRNA, translation, and protein synthesis. The third section focuses on the physiological functions of the human body (e.g., cardiovascular system, nervous system, etc.).

COURSE CONTENT: The course in molecular and cellular biology is organized according to the following concepts, in which molecular and cellular mechanisms are addressed from a comprehensive perspective: Basic unit of life; Membrane-bound structure; Genetic material Metabolism; Reproduction; Response to stimuli; Homeostasis; Division of labor (energy) in multicellular organisms; Evolutionary adaptability.
The role of these various mechanisms is then illustrated in the context of molecular and cellular deregulations that occur in mammary carcinogenesis (breast cancer).
The anatomy/physiology course presents the main vital functions of the human body. Anatomy and physiological functions are presented according to the following outline: Introduction; Origins; Integumentary system; Skeletal system; Muscular system; Nervous system; Cardiovascular system and blood; Respiratory system; Digestive system; Endocrine system; Excretory-urinary system; Immune system; Reproductive system; Conclusions for the engineer
ASSESSMENT METHOD: Written exam and report.

Course Name: DNA and microfluidic practical work (processing of bioelectric signals for E-Health)

PREREQUISITE: Knowledge in electronics; Knowledge in bioelectricity; Knowledge of microcontrollers (STM32); Python for AI-based classification
LEARNING OUTCOMES: This lab focuses on designing a biomedical system (EMG, impedance spectroscopy, or custom project), from literature review to AI-based data analysis. Students develop a microcontroller-based solution, perform testing, collect data, and deliver a final report.
COURSE CONTENT: 1) Selection of a topic from the following options: Develop an EMG (electromyogram) system for muscle activity, create a biometric system using impedance spectroscopy. Literature review and implementation principles. Design the system within the framework of microcontroller-based development. Testing and data collection. Data analysis and AI-based classification.
ASSESSMENT METHOD: Practical work and project.

Course Name: Anatomy

Syllabus pending

MODULE: INTRODUCTION CYBER

Course Name: Fundamentals of Cyber

PREREQUISITE: Fundamentals on operating systems, proficiency with Linux command-line tools, basic web development knowledge, development C/C++, development scripting shell

LEARNING OUTCOMES: During this session, we explore the history of network and computer hacking and analysis some of the biggest attacks having occurred in the past decades. Through these stories we introduce the cyber security vocabulary, study what cyber threat actors are and the kind of vulnerabilities they are commonly exploiting, and how to protect yourself with good computer hygiene practices. Finally, students explore their own hacking capabilities with first hands-on labs.

COURSE CONTENT: This introductory cybersecurity course is organized into three progressive chapters that are built from foundational theory and best practices to hands-on technical skills:

The first chapter provides a foundational deep dive into cybersecurity's origins, core concepts, and essential hygiene practices. Students begin by tracing over seven decades of cyber evolution, from the earliest communication networks to today's AI-driven threat landscape, while learning the vocabulary and tactics that have shaped modern defense frameworks. The emergence and impact of Dark Web networks are then explored, revealing how anonymity technologies fuel both legitimate and illicit activities. Next, participants master information-security fundamentals (CIA/AAA models, NIST and MITRE frameworks, vulnerability lifecycles, risk management, DevSecOps) alongside practical "cyber hygiene" techniques: identity and access management (password best practices, MFA, passkeys), data encryption at rest and in transit, endpoint hardening, VPN usage, and social-engineering defenses. Throughout this chapter, hands-on Linux labs reinforce each concept, guiding students through real-world misconfigurations, scripting errors, and firstline forensic investigations.

The second chapter immerses students in the evolution and security of web protocols before guiding them through practical auditing techniques. Beginning with the history and mechanics of HTTP—from version 1.0 through the performance and feature enhancements of HTTP/2 and HTTP/3—participants examine HTTP message structure, multiplexing, header compression, and the role of proxies. An in-depth study of HTTPS then unpacks TLS handshakes across versions 1.0 to 1.3, perfect forward secrecy, cipher-suite negotiation, and the limitations of PKI and DNS-based certificate validation. Building on this foundation, the course introduces web-application security testing: students learn to configure intercepting proxies, perform vulnerability scans and manual tests, and execute OWASP Top 10 exploits (SQLi, XSS, CSRF, LFI, SSRF, IDOR, etc.) in dedicated labs. "Security by design" principles and mitigation strategies are woven throughout. By chapter's end, learners will have both a deep theoretical understanding of web protocol security and hands-on experience in designing, testing, and hardening real-world web applications.

The third chapter dives straight into low-level exploitation and memory-based vulnerabilities, building on the protocol and hygiene foundations of earlier chapters. Students begin by mastering CPU and memory fundamentals, understanding x86/x64 memory layout (stack vs. heap), calling conventions, and low-level data representations. The core of the chapter focuses on buffer-overflow attacks: from the theory of stack smashing and return-to-libc to advanced techniques such as ROP chain construction. Participants then study modern

mitigation mechanisms (ASLR, DEP/NX, stack canaries) and learn how to bypass or reinforce them. Through hands-on labs, they craft and deploy working overflow exploits against simple C/C++ targets, experiment with fault injections, and implement countermeasures to harden applications. Students will grasp how memory corruption underpins real-world attacks and how layered defenses can detect, prevent, and mitigate these critical threats.
The cybersecurity introduction course equips students with the vocabulary, methodologies, and hands-on experience needed to navigate today's threat landscape, from understanding why attacks happen, to testing and hardening web applications, to dissecting the very building blocks of software at the machine level.
ASSESSMENT METHOD: Practical work and written exam.

Course Name: Cryptography Level 1

PREREQUISITE: Fundamental principles of cybersecurity: CIA properties, AAA models, principle of least privilege and secure-by-default, defense in depth, Zero Trust model, vulnerability lifecycle management, understanding the exploit chain, and risk management. Maths for engineers (modular arithmetic, finite fields). High-level programming language (e.g. Java).
LEARNING OUTCOMES: This course introduces the basics of cryptography, including the fundamental principles of encryption, hashing, and integrity algorithms. Students will learn to understand the role of these techniques in protecting information and recognize the principles underlying simple cryptographic systems.
COURSE CONTENT: Chapter 1: Introduction to Cryptography. Chapter 2: Encryption Algorithms. Chapter 3: Hash Functions and Integrity. Chapter 4: Digital Signatures. Chapter 5: Key Exchange. Chapter 6: Introduction to Public Key Infrastructure (PKI)
ASSESSMENT METHOD: Written exam and project.

MODULE: COMMON MODULES

Course Name: Technical Project

PREREQUISITE: Basic knowledge of project management & technical course related to the project topic.

LEARNING OUTCOMES: This technical project aims to enable students to apply the knowledge acquired (or currently being acquired) in their chosen specialization through the completion of a comprehensive project. It is designed to develop both specific technical skills and cross-disciplinary project management skills, such as planning, coordination, teamwork, and communication. Supervised by the specialization advisor, the project prepares students for managing complex projects in a professional context.

COURSE CONTENT: Needs analysis, state of the art, feasibility, design, validation. Architecture, development, technological choices. Prototyping, testing, documentation (document management and writing). Written reports, oral presentation, communication to a non-expert audience. Time management, quality, costs, ethics. Project management, teamwork, shared tools.

ASSESSMENT METHOD: Project, report + defense by group.

Module: SHES (Social, Human and Economical Science)

Course Name: Agility Project Management

PREREQUISITE: Know the key concepts of project management (scope, stakeholders, objectives, costs and deadlines).

LEARNING OUTCOMES: Understand Agile philosophy, get first experience, understand the SCRUM methodology within an Agile context.

COURSE CONTENT: Discover Agility thanks to a learning experience during a Lego for SCRUM
Understand main Agile notions through SCRUM methodology
- Sprint
- Backlog
- MVP/MMP
- US/DOR/DOD
- Velocity
- Relative estimation
- Ceremonies and retrospective

ASSESSMENT METHOD: Assessment through exams

Course Name: Strategy and development of Startups

PREREQUISITE: Basics of how businesses work and its main functions.

LEARNING OUTCOMES: This course introduces the themes and the main concepts related to entrepreneurship or the creation of a startup.

COURSE CONTENT: Introduction: Type of company, articles of association, associated agreements and intellectual property law. Market research. Strategic Plan: Business, Marketing & Communication, Writing the Deck, Preparation of the DMaison (Pitch), Sources and Financing Methods, Support of your creative project

ASSESSMENT METHOD: Work at home and written exam

Course Name: Professional Personal Preparation

PREREQUISITE: To have already worked before the course on their CV and created a LinkedIn account.
LEARNING OUTCOMES: Prepare for recruitment processes. Have an impactful LinkedIn account and an easy recruitment CV. Be able to drive an interview and present themselves in a concise manner (pitch presentation).
COURSE CONTENT: Impeccable CV
Impactful and attractive LinkedIn account
Present self in a synthetic manner
Learn how to drive an interview
Search for companies - how to find them et contact them
ASSESSMENT METHOD: Practical work

Course Name: Research Methods

PREREQUISITE: No specific prerequisites.
LEARNING OUTCOMES: Prepare for recruitment processes. Have an impactful LinkedIn account and an easy recruitment CV. Be able to drive an interview and present themselves in a concise manner (pitch presentation).
COURSE CONTENT: Impeccable CV
Impactful and attractive LinkedIn account
Present self in a synthetic manner
Learn how to drive an interview
Search for companies - how to find them et contact them
ASSESSMENT METHOD: Practical work

Course Name: Problem Resolution

PREREQUISITE: No specific prerequisites.
LEARNING OUTCOMES: Provide students with the right processes to enable them to successfully complete their problem solving.
Know how to use the right tools at the right time and in the right way.
COURSE CONTENT: How to amplify a problem? How do you manage not to solve a problem? Different ways of dealing with a problem. How to pose and solve a problem? How do I define a goal? (SMART, problem-solving tools, SWOT, Brainstorming, CQQCOQP, Ishikawa, Pareto diagram – 20/80, Impact Feasibility, Compatibility Matrix, Stern's Cube, Mind mapping) and problem-solving games.
ASSESSMENT METHOD: Work at home

Language Courses

LANGUAGES AVAILABLE FOR INTERNATIONAL STUDENTS:
French for foreign students
English