Classes in English at Polytech IG – 2023/2024

Fall Semester

Computer Science and Management (IG department)

Courses (fall semester)	***Teacher(s)***	***Hours***	***Credits***
F1. Software Engineering and Design Principles (fourth year)	*T. Stratulat*	*60*	*3,5*
F2. Data warehouse and reporting (fourth year)	*A.Laurent*	*24*	*1.5*
F3. Machine Learning and Data Mining (fourth year)	*A.Laurent*	*24*	*1,5*
F4. Web APp Engineering 2 (WAPE 2) (fifth year)	*C. Tibermacine*	*30*	*2*
F5. Project of Web APp Engineering 2 (P-WAPE 2) (fifth year)	*C. Tibermacine*	*12*	*2*
F6. Recommendation Systems (fourth year)	*E. Pacitti*	*5*	*0,5*
F7. Software Engineering Practices (fourth year)	*T. Stratulat*	*24*	*2,0*
F8. Industrial project (fourth year)	*L.Buisson-Lopez*	*35h/week during 8 weeks*	*5*
F9. Artificial Intelligence & Multi-Agent Systems	*T. Stratulat*	*24h*	*2*

F1 . Object-Oriented Software Engineering (fourth year)

Long description:

The aim of this course is to provide students with the skills needed to work on large-scale IT projects characterized by a high degree of complexity in terms of customer requirements, functionalities, lines of code, people involved in the project lifecycle, budget and time constraints, etc. Students will therefore first be confronted with general concepts and principles that will guide them in analyzing, specifying and designing high-quality, easily modifiable software in a classic development context. Students will therefore be introduced to general concepts and principles that will guide them in analyzing, specifying and designing high-quality, easily modifiable software in a classic (V-cycle) or agile development context. Then, through complex examples, they will be exposed to best practices for adding new functionalities while preserving

software quality.

The content, structure and practices of this course are designed to help students understand that to be a computer engineer it is to be not only just a programmer, but also an architect, an analyst, a tester, a manager, and so on.

This course covers the following topics:
1) Object-oriented design and programming from a software engineering perspective.
- Revisiting the concepts of: abstraction, encapsulation, interfaces, inheritance, polymorphism, reuse, delegation.
- SOLID design principles (Single Responsibility, Open-Close, Liskov Substitution, Interface Segregation, Dependency Inversion)
2) UML modeling and tools for object-oriented and architectural design
3) Modular software design: divide and reign, Low-coupling, High-cohesion, Increased abstraction, reuse, etc.
4) Reuse and reusability, generic software design and framework-based software design.
5) Design Patterns: Singleton, Factory Method, Abstract Factory, Composite, Façade, Proxy, Adapter, Delegation, Observer, DAO, Immutable, etc.).
6) Architectural models: multi-level (n-tier), distributed, client-server, pipeline, P2P, broker, transactional, message-oriented, service-oriented.

Prerequisites: An introductory course on an object-oriented programming language is preferable but not mandatory.
Evaluation: Written exam
Contact: tiberiu.stratulat@umontpellier.fr

F2. Data Warehouse and Reporting (fourth year)
(booklets/slides in english, spoken language in class to be confirmed) This course introduces the core concept of data warehouses by presenting the main differences between operational (OLTP) and decisional (OLAP) databases. The multidimensional model and key performance indicators are presented together with the concept of ETL. Professional tools are used during hands-on tutorials.
Prerequites : This course requires prior knowledge in SQL.
Evaluation : The examination consists of a written test and a report of hands-on sessions.
Contact : Anne.Laurent@lirmm.fr

F3. Data Mining and Machine Learning (fourth year)
(booklets/slides in english, spoken language in class to be confirmed) This course presents the main methods of data mining from the computer science perspective: supervised and unsupervised algorithms such as decision trees, naive Bayes, k-nearest neighbours, k-means,… and pattern mining (frequent item-sets, association rules, sequential patterns). The course also focuses on evaluation methods (confusion matrix, quality measures).
Prerequites : No prior knowledge required for this course.
Evaluation : The examination consists of a written test.
Contact : Anne.Laurent@lirmm.fr

F4. Web APp Engineering 2 (WAPE 2) (fifth year)
(only slides in english)
The goal of this module is to provide students with an introduction to advanced languages and frameworks for Web and reactive application development in the Java ecosystem, like JEE microservices, JPA, Spring MVC and Spring WebFlux. We also introduce the specifities of frameworks in other ecosystems like ASP.NET and Scala Play. In addition, in this module we address Cloud deployment of such kind of applications.
Prerequisites: introductory course on Web App Development

Evaluation: quiz and lab work
Contact: Chouki.Tibermacine@umontpellier.fr

F5. Project of Web APp Engineering 2 (P-WAPE 2) (fifth year)
The goal of this project is to develop and deploy a real-world Web reactive application with Spring framework.
Prerequisites: Web APp Engineering 2 (WAPE 2)
Evaluation: project deliverables and final oral presentation & demo
Contact: Chouki.Tibermacine@umontpellier.fr

F6. Recommendation Systems (fourth year)
This cours introduces the fundamental concepts of recommendations systems and the most important models : collaborative filtering, content-based filtering, social based filtering. The top-N recommendation algorithm is presented and experienced with a pratical exercice.
Prerequites : familiarity with matrix computation, and java language
Evaluation : A report
Contact : Esther.Pacitti@lirmm.fr

F7. Software Engineering Project (fourth year)
The goal of this module is to provide students the opportunity to practice and use the appropriate tools to develop software projects using a systematic approach such as the V-model, modular and object-oriented design, and object-oriented technologies.
The subject is either provided by an industrial partner or proposed by the students. 4 or 5 students compose project teams, and they take on responsibilities in turn. At each phase, a student should consider working at least 3 use cases, different from those of the previous phase, ultimately covering all use-cases and project phases equally.
Prerequisites: Participation in the Object-Oriented Software Engineering course (F1) or previous project experience using object-oriented design or programming language.
Evaluation: The project is evaluated according to the quality of deliverables that students provide at each phase (reports, UML diagrams, use case reports, tests, code, user interface design, database design, etc.), and the management activities they involve in (roles, meeting facilitation, initiative, creativity, etc.).
Contact: tiberiu.stratulat@umontpellier.fr

F8. Industrial Project (fifth year)
Team project, practicals, design and development of information systems to meet the real demands of industrial partner companies or clients. Students learn to conduct an IT project in a professional context and face real difficulties.
Context & Requirements: students are supervised by a school tutor who verifies the implementation of good practices in project management methodology. Teams of students can also ask any member of the teaching team for their technical expertise. Each team has a personal project room with the equipment necessary to conduct the project.
Prerequisites: project management methodology and technical skills in IT design & development but above all responsibility, autonomy and communication.
Evaluation:
Regular monitoring of intermediate deliverables by the school tutor
4 final deliverables: a summery report – a technical report – a scientific poster – an oral defense
Contact : lysiane.lopez@umontpellier.fr

F9. Artificial Intelligence & Multi-Agent Systems

Short description: This course is an introduction to the large field of Artificial Intelligence.

Long description. This is an introductory course of AI. As the general field of AI is very broad, the purpose of this course is to expose students to some subareas such the agent-oriented paradigm, search agents, logical agents, and reactive multi-agents. The students are supposed to be able to know how to characterize an AI problem and its environment. After this course they could have the prerequisites to focus on other AI subareas such as Deep Learning and Generative AI, which are studied in other courses.

This course covers the following topics.
1. Introduction: history of AI, definition, subdomains, problems, current trends
2. AI and agents: PEAS, environments, types of agent architectures
3. Search Agents:
- uniformed search: breadth-first, depth-first, depth limited, iterative-deepening, uniform-cost, etc.
- informed and local search: greedy, A, hill climbing, genetic algorithms*
4. Knowledge based agents:
- representation and reasoning.
- logics (propositional and first order logic) inference, entailment, model checking, resolution, forward chaining, backward chaining.
5. Multi-agent systems
- agent architectures
- reactive agents: FSM, situated actions, Subsumption.
- rational agents: BDI, intentional stance, modal logic
- social agents: communication, commitments, protocols
6. Netlogo, Madkit programming

During the laboratories the students use Netlogo to implement various kinds of agents in a specific environment.
Textbook: Artificial Intelligence: A Modern Approach, by Peter Norvig and Stuart Russell, Pearson, 2021
Prerequisites: Basic notions of algorithms.
Evaluation: The final grade is obtained on the basis of a written examination 50% and laboratory work 50%.
*Contact: **tiberiu.stratulat@umontpellier.fr***

Spring Semester

Computer Science and Management (IG department)

Courses (spring semester)	***Teacher(s)***	***Hours***	***Credits***
S1. Object oriented design and programming (lecture+practical) (third year)	*C. Tibermacine*	*25*	*3,5*
S2. Industrial project (fourth year)	*L. Buisson-Lopez*	*35h/week during 8 weeks*	*5*

S1. Object-Oriented Design and Programming (third year)

Fundamental concepts of object-oriented programming (OOP) and design are presented: encapsulation, inheritance, interfaces, polymorphism, abstract data type, as well as good and bad practices in the field. The concepts are applied in the Java language through a set of labs tightly intermixed with courses. Specific aspects of Java that can be found in other OOP languages are also evoked, such as exceptions, libraries and input/output operations. OOP is practiced at first in standalone and online environments dedicated to Java (masking some difficulties), then on the command line (to ensure a fine understanding of the compilation and execution mechanisms), and finally in an integrated development environment (namely, Eclipse). Aiming to bring students at an operational level, the course requires students, to work individually or in teams depending on the labs, to submit milestones according to deadlines, and to collaborate through both a LIMS and a distributed revision control system (namely, git).

Prerequisites: the students are assumed to know how to code in at least one programming language such as Python, C, PHP, Javascript, ...(HTML being not enough). Prior to this course they must understand basic concepts such as variables, types, basic data structures, and must have knowledge of basic flow-control instructions (conditionals, loops, functions, ...).

No prior experience of the Java programming language nor of the UML formalism is assumed.

Evaluation: short quizzes are proposed at different moments along the course, some labs will be evaluated (sometimes with extra times to be submitted). A final written exam on paper closes the training module.

Contact: chouki.tibermacine@umontpellier.fr

S2. Industrial Project (fourth year)

Team project, practicals, design and development of information systems to meet the real demands of industrial partner companies or clients. Students learn to conduct an IT project in a professional context and face real difficulties.

Context & Requirements: students are supervised by a school tutor who verifies the implementation of good practices in project management methodology. Teams of students can also ask any member of the teaching team for their technical expertise. Each team has a personal project room with the equipment necessary to conduct the project.

Prerequisites: project management methodology and technical skills in IT design & development but above all responsibility, autonomy and communication.

Evaluation:

Regular monitoring of intermediate deliverables by the school tutor

4 final deliverables: a summery report – a technical report – a scientific poster – an oral defense

Contacts : lysiane.lopez@umontpellier.fr