ERASMUS MUNDUS JOINT MSC IN ADVANCED SYSTEMS DEPENDABILITY (DEPEND)

MODULE LEARNING OUTCOMES[i]

Module Learning Outcomes

Module	**Short Title**	**Learning Outcomes** **On successful completion of the module, students should be able to:**
EMJM1	RIGOROUS SOFTWARE PROCESS	Reason about why software projects fail. Identify the role of rigor in the software process in improving the chances of success. Create mathematically precise specifications and designs using languages such as OCL AND B. Analyse the properties of formal specifications and designs, performing proofs of correctness. Analyse the correctness of object-oriented programs (eg. static analysis, simulation, model checking). Describe how tools that assist in this analysis are designed and implemented.
EMJM2	REQUIREMENTS ENGINEERING & SYSTEM	Select and apply various requirements elicitation techniques. Document, validate and track any changing requirements. Model various aspects of software systems using each of the diagrams of UML. Choose and apply a selection of software design patterns. Describe implement strategies for a number of design patterns.
EMJM3	TESTING	Describe the principles of software testing; Describe and compare a number of standard testing techniques for unit and system testing; Select test techniques and design test cases; Design and implement automated tests; Evaluate test coverage; Describe and evaluate the relevance of testing in the software development process.
EMJM4	DIRECTED READING	Be able to critically analyse research and other publications; be able to summarise the content of a research paper and produce an abstract; be able to critically review a research paper and assess it for strengths and weaknesses; be able to produce an effective poster.
EMJC1	JOINT ACTIVITIES	Be able to apply an up-to date knowledge of state of the art in selected areas of dependability to their work.

EMJC2	OO PROGRAMMING	Construct object oriented programming solutions for reuse, using abstract data types that incorporate encapsulation, data abstraction, information hiding, and the separation of behavior and implementation. Construct multiple-file or multiple-module programming solutions that use classes, subclasses, class hierarchies, inheritance, parameterised classes, generics and polymorphism to reuse existing design and code. Compare and contrast a range of object-oriented programming languages. Create programming solutions that use data structures and existing libraries. Perform object-oriented design and programming with a high level of proficiency in more than one object-oriented programming language.
EMJC10	SUMMER SCHOOL (YEAR 1)	Be able to describe their work through a formal presentation and a poster presentation; be able to ask valuable questions; have experienced the culture of another country.
EMJC11	SUMMER SCHOOL (YEAR 2)	Be able to describe their work through a formal presentation and a poster presentation (in an advanced way); be able to ask valuable questions; have experienced the culture of another country.
EMJM10	MATHS OF CS	Students will apply an understanding of the fundamental and practical limits of computers.
EMJM11	CRYPTOGRAPHY	Apply an understanding of the concepts, implementation, and deployment of practical cryptography.
EMJM12	INTERACTION DESIGN	Describe the main features of interaction design as applied to computer systems. Analyse trends in computer hardware and software that have implications for future modes of interaction. Construct low-fidelity interface prototypes. Implement a simple user interface on a mobile computing device. Recognise strengths and weaknesses in particular interaction designs. Design an interface that adheres to good interaction design principles.
EMJM13	PROGRAM COMPREHENSION	Describe and compare approaches that aid the comprehension of object-oriented software. Categorise the kinds of information that can be extracted from a program using different analysis techniques, giving examples. Choose appropriate models and representations for given program comprehension tasks. Propose and design solutions to program comprehension tasks in the context of software engineering.
EMJM14	INTERNET SOLUTIONS	Be able to apply an understanding in the development of multi-tiered, server-side, web-based software applications, web standards, client and server-side programming and database integration.
EMJM15	DATABASES	Design and implement spatial databases using standard models and spatial database management systems Analyse and optimise spatial database designs to maximise efficiency and effectiveness Query spatial databases using standard query tools and languages Create interfaces to view and customise, interact with spatial data Design and implement spatial indices for efficient searching of data Ensure reliability, security, integrity and privacy in spatial databases

EMJM16	MINOR DISSERTATION	Conduct a concise investigation into a topic of interest, importance or relevance to software engineering or the application of software engineering principles to some aspect of your work placement. Work largely independently, within agreed project requirements, with minimal supervision. Complete an abbreviated project report covering central aspects of the project and make a presentation based on the project. Work within all project deadlines, consultation requirements and give regular feedback to your supervisor(s). Work effectively on a challenging topic. Research alternative solutions. Develop and evaluate a solution, and to evaluate the process that lead to the solution.
EMJM50	PLACEMENT	Taking direction on work placements from the work placement officers, meeting all deadlines set by them and any potential employers. Finding a suitable work placement by participating in suitable application and interview processes. Participating in a place of work in a manner that is agreeable to the employer, and making a contribution. Writing a work placement report, documenting their main contributions to the place of work, and reflecting on what they have learned through their work experience.
EMJM51	PROJECT & DISSERTATION	Conduct a reasonably thorough investigation into a topic of interest, importance or relevance to software engineering or the application of software engineering principles to some aspect of your work placement. Work largely independently, within agreed project requirements, with minimal supervision. Complete a detailed project report covering all central aspects of the project and make a presentation based on the project. Work within all project deadlines, consultation requirements and give regular feedback to your supervisor(s). Work effectively on a challenging topic. Research alternative solutions. Develop and evaluate a solution, and to evaluate the process that lead to the solution.
EMJS10	SOFTWARE ENGINEERING PRINCIPLES	Understand the key concerns that are common to all software development processes. Be able to select appropriate process models, approaches and techniques to manage a given software development process. Be able to elicit requirements for a software product and translate these into a documented design. Be able to identify dependability and security issues that affect a given software product. Understand the role that testing and reuse play in the implementation phase and how these activities relate to the wider software process.
EMJS11	SOFTWARE ENGINEERING PRACTICE	Be able to translate a set of requirements and a high-level design into a complex working system. Be able to develop the system implementation using the test-driven development approach. Be able to write tests for a complex system consisting of a set of collaborating classes. Be able to reuse components and access remote services. Be able to utilise version control and build management systems to manage a ollaborative implementation process.
EMJS12	SOFTWARE ARCHITECTURE	Be aware of the key elements of software architecture. Be familiar with a variety of architectural styles and how they may be combined in a single system. Have a working knowledge of software architecture design for a non-trivial system. Understand how software architecture aids different stages of the software lifecycle.
EMJS13	CRITCAL SYSTEMS ENGINEERING	Understand the fundamental notion of system dependability and how dependability is affected by both social and technical factors. Understand the notion of a socio-technical system. Have been introduced to methods and techniques for achieving software dependability. Understand how human and social factors may have both positive and negative influences on socio-technical system dependability.

EMJS21	AI PRINCIPLES	Be aware of how search is used to solve a variety of problems in AI. Be familiar with fundamentals of symbolic AI, machine learning, neural networks, and robotics.
EMJS22	AI PRACTICE	Know how search is used in practice to solve a variety of problems in AI. Be able to implement fundamental techniques of optimisation, reasoning with uncertainty, logic and knowledge representation, and AI search
EMJS23	LANGUAGE & COMPUTATION	Have an understanding of modern approaches to automatic natural language processing (NLP).Have an understanding of statistical and symbolic approaches to model human language processing. Be able to build (parts of) small NLP applications
EMJS31	DATA-INTENSIVE SYSTEMS	Be able to engineer and work with systems which need to process big data
EMJS32	KNOWLEDGE DISCOVERY & DATA	Understand historical/philosophical perspectives, and be able to model selection algorithms and optimality measures, tree methods, bagging and boosting, neural nets, and classification in general, and apply in practical applications (using R, SAS or python).
EMJS1	LOGIC AND SOFTWARE VERIFICATION	Be aware of the need for formal methods and software verification. Be familiar with a variety of well known model checkers. Have a working knowledge of relevant approaches to modelling and property specification. Understand the fundamental principles underlying model checking. Be able to use transition systems as a model for systems, and temporal logics to express system properties including functional correctness, reachability, safety, liveness, fairness and real-time properties. Be able to specify and verify models using various approaches including PROMELA and SPIN, timed Automata and UPPAAL (real-time systems), and Petri nets
EMJS2	HCI PRINCIPLES AND METHODS	Be aware of how the human interacts with computers in a variety of modern computing scenarios. Be familiar with fundamentals of evaluation, usability, cognition, ergonomics, computing paradigms and heuristics in HCI
EMJS3	INTERACTIVE SW AND HW	Be able to create basic interfaces with a range of new technologies. Have an understanding of how interface implementation is organised. Have an improved ability to get started with new technologies quickly.
EMJS4	INFORMATION VISUALISATION	Be able to design and justify the design of a visual representation of a given data set. Be able to critically assess the design of a data visualisation based on the nature of its underlying data, targeted audience, and general purpose. Be able to make use of existing low- and high-level visualization toolkits to create effective information visualisations. Be able to communicate and document their visualisation design process and outcomes.
EMJS5	ADVANCED NETWORKS	Have an in-depth understanding of at least one advanced topic in computer networking. Be familiar with key networking abstractions and concepts. Have a working knowledge of common network tools and monitoring software. Be able to analyse traffic traces. Be familiar with current networking technologies.

EMJS6	MOBILE & WIRELESS	Have knowledge and understanding of basic mobile network architecture. Have knowledge and understanding of some basic technologies that are in use. Be able to make critical assessment of mobile systems. Be able to analyse and propose broad solutions for a range of mobile scenarios.
EMJS7	CONCURRENCY	Be aware of the need for abstraction in parallel programming. Be familiar with a variety of common high-level patterns of parallel programming. Have a working knowledge of the implementation of those patterns using Haskell. Understand performance and debugging issues on both CPUs and special-purpose parallel hardware.
EMJS50	PROJECT & DISSERTATION	Write an individually supervised dissertation, on a topic in computer science, comprising a literature review, extension of old or development of new ideas, their implementation and testing, summarised in a report, with the implementation based on sound theory and software engineering principles, and give a presentation of their work.
EMJS51	INDUSTRIAL RESEARCH PLACEMENT AND SUPPORT MODULES	Execute an in depth research study, study advanced theoretical concepts, design and setting up of experiments, analysis of measurements, etc.
EMJL1	ADV SOFTWARE ENGINEERING	Apply advanced software engineering techniques to the development of dependable software.
EMJL2	NON CLASSICAL LOGIC & PROOFS	Execute proofs using non-classical logic, with application to software systems.
EMJL3	SOFTWARE MODELLING	Be able to construct and evaluate mathematical models of software systems.
EMJL4	PROOF OF ALGORITHMS	Understand the best techniques to apply, and be able to prove the correctness, or otherwise, of algorithms.
EMJL5	ALGORITHMIC VERIFICATION	Understand the best techniques to apply, and be able to verify the correctness, or otherwise, of algorithms.
EMJL6	PROJECT MANAGEMENT	Gain the tools to manage a project; identify the stages involved in project planning; use a Gantt chart (or equivalent) with milestones to monitoring and measuring progress taking account of budget and risks.
EMJL7	INTEGRATION PROJECT	Design and implement a small-scale interaction program.

EMJL20	ADVANCED PROOF OF PROGRA	Be able to apply advanced proof techniques in proving the correctness of programs.
EMJL21	SEMANTICS, PROOFS & TYPES	Be able to apply an understanding of semantics, proofs, and types in developing proofs and related activities.
EMJL22	DECISION PROCEDURES FOR VERIFICATION	Understand and be able to apply decision procedures for program verification.
EMJL23	REWRITING FOR PROGRAM PROOVING	Be able to rewrite/restructure a program to a form in which program proving techniques can be readily applied.
EMJL24	DATA ENGINEERING	Understand and be able to apply the principles for analysing large data sets (Big Data).
EMJL25	PROTOCOL SECURITY	Be able to apply mathematical techniques to investigate the security of protocols.
EMJL26	ADV CRYPTOGRAPHY	Be able to describe and use advanced cryptographic approaches.
EMJL27	MODELS OF COMPUTATION	Be able to demonstrate a knowledge of different models of computation, and know when to apply them.
EMJL28	MSC OPTION	An option from the MSc catalogue (with the approval of the Local Course Directors in Nancy)
EMJL29	PROFESSIONAL MODULE	Demonstrate an understanding or professional aspects of the work place and expected competences of an engineer.
EMJL30	LANGUAGE MODULE: FRENCH	Develop a working knowledge of the French language, with special focus on technical French for Computer Science and Dependability.

EMJL51	INTERNSHIP/RESEARCH PROJECT	During the year in UL the students, during the second semester an internship project (4-5 months) inside one of the teams of the LORIA laboratory, will have experience in, and be able to conduct research in, topics that cover different strong themes of the laboratory: Algorithms, Geometry and Image; Formal Methods; Networks, Systems and Services; Natural Language Processing and Knowledge Discovery; Complex Systems, Artificial Intelligence and Robotics. In addition, the students will have developed skills in working in a professional research laboratory, and be well equipped for work after the course.

[i] We have done our best to ensure the accuracy of the information shown here, but we accept no responsibility or liability for any incorrect material.