A. core courses		
SWE 540	**Research Methods in Software Engineering**	**3 (3+0)**
This course introduces different research methods as well as some of the statistical, modeling, and optimization techniques used in software engineering research. Research methods covered include empirical, semi-formal, and formal approaches. Statistical sampling and analysis and its application to software engineering problems will be a key part of the course. Key modeling and optimization techniques using mathematical logic, graph theory, and operations research will be addressed as well. Students will review different documents, case studies, and research papers to have a practical understanding of the use of the various research approaches in modeling, analyzing, and solving relevant software engineering research problems.		
SWE 541	**Software Modeling**	**3 (3+0)**
Models are the basis for the analysis and specification of software artifacts. This course exposes the students to a detailed description of different software models at different abstraction levels along with their purposes, the context in which they are used, and the operations that can be applied on them. Topics covered include an overview of modeling in general and in the context of software development – modeling software behavior: flowchart, decision tables, Petri nets; Unified Modeling Language (UML); Metamodeling: Meta-Object Facility (MOF); XML/XMI, UML extensions: profiles; models transformation: Query/View/Transformations (QVT), and the Atlas Transformation Language (ATL). A brief overview on some formal languages/methods (OCL, Z, B, etc.) will be discussed. The course will also discuss aspect-oriented modeling as well as model-driven development (MDD). The student will be exposed to the research literature relevant to these topics.		
SWE 543	**Software Requirements Engineering**	**3 (3+0)**
This course will cover tools, notations, and validation techniques for the analysis, specification, prototyping, and maintenance of software requirements. It will include in the beginning a review of the main techniques and models used in requirements Engineering including elicitation, evaluation, specification and documentation, prioritization, traceability, quality assurance, evolution. Then the course will cover aspects related to building system models for requirements engineering; Modeling system objectives with goal diagrams; risk analysis on goal models; modeling conceptual objects with class diagrams; modeling system agents and responsibilities; modeling system operations; modeling system behaviors; integrating multiple system views; reasoning about system models. Students will be exposed to the research literature related to requirements engineering.		
SWE 542	**Software Design and Architecture**	**3 (3+0)**

	This course exposes students to concepts and methods for the architectural design of large-scale software systems. Topics covered in this course include fundamental design concepts, design patterns, anti-patterns and their application. The course will also cover design aspects relevant to non-functional requirements such as availability, interoperability, modifiability, performance, security, testability, usability. Advanced topics such as ADL (architecture description language) and xADL will be covered. Methods related to the evaluation of software architectures (such as the Architecture Tradeoff Analysis Method (ATAM)) will be discussed in order to compare and contrast the components, connections, protocols, topologies, constraints, tradeoffs, and variations of different types of architectural styles. Students will be exposed to the research literature relevant to software design and architecture.	
SWE 545	**Software Testing and Quality**	**3 (3+0)**
	This course is about Software Quality Assurance and Testing, a key knowledge area in software engineering. On the quality part, the course covers software quality factors, errors, faults and failures, software quality reviews and inspections, software quality metrics, and software quality standards. In the testing part, the course covers testing processes, test management, test techniques: static techniques (review, static analysis), dynamic techniques (black-box, white-box, defect-based, dynamic analysis), test of software characteristics. Test automation will be also discussed. Recent advancements in testing such as crowd testing, test-driven development (TDD), etc. will also be covered. Reading and presenting relevant research papers is an important part of the course.	
SWE 546	**Software Processes and Management**	**3 (3+0)**
	This course covers two important knowledge areas in software engineering: software processes and software project management. Topics related to software processes include software process modeling, prescriptive process models, and descriptive process models. Software development methodologies including waterfall-like, spiral, prototyping-based, adaptive methodologies, as well as agile methodologies such as scrum and Kanban will be also reviewed. Topics related to software project management include Scope, Time, Cost, Quality, Risk, Communications, Human Resources, stakeholder, and Integration Management. Relevant research problems will be discussed including issues related to software process improvement, staffing and scheduling, effort and cost estimation, best and bad practices in project management, etc. Reading and presenting relevant research papers in software processes as well as in software project management will be an important part of the course.	

SWE 596	Thesis Proposal Preparation	one study unit
The student enrolled in the thesis option prepares an MSc thesis proposal that includes the research problem(s) to be addressed by the proposed research, a thorough literature review of the related works, the objectives, the methodology to be followed, the results and contributions expected from the proposed research, as well as timeline and schedule of the proposed research. The research proposal will be evaluated according the university regulations and college/department internal procedures.		
SWE 594	**Graduation Project Preparation Study**	**one study unit**
This 1 study unit course is a preparation study for the project that shall be done by students as a pre-requisite for the project. Students work closely with the supervisor to define the scope of the project and understand its requirements, identify the tools required to do the project, and review relevant related literature. The student shall submit a written report to his supervisor at the end of the semester.		
SWE 595	**Graduation Project**	**(4) study units**
During their study in the program, a student acquired new knowledge and skills or developed what they already have. Following the project preparation study, this course gives the student the opportunity to use this knowledge and these skills in a project of appropriate complexity. The project can take the form of a theoretical or experimental study (analysis, evaluation, comparison, etc.) or the design and/or implementation and/or maintenance of one or more components of a system. Students write a report describing their work and perform an oral presentation in front of an examination committee.		
SWE 600	**Thesis**	**(6) study units**
The student writes a detailed thesis dissertation report describing in details all aspects of the research work accomplished and defends publicly his work in front of a defense committee according to the university regulations and college/department internal procedures.		
B. Elective Courses		
SWE 544	**Distributed Software Systems**	**3 (3+0)**
Almost every large system is a distributed system. This course focuses on engineering such systems with an emphasis on what makes them different. Covered topics include the following: the characteristics of distributed systems; distributed systems models; inter-process communication and the client-server architecture; remote invocation; publish/subscribe systems; message queues; shared memory; web services; peer-to-peer systems. The course contains laboratory work that will allow the students acquire hands-on experience in implementing distributed systems. The student will be exposed to the research literature relevant to these topics.		

SWE 547	Formal Software Specification and Verification	3 (3+0)
This course is concerned with mathematically rigorous techniques and tools for specifying, validating, and verifying software systems. The specification of software deals with using well-formed statements expressed in formal languages (such as Z, B, and OCL) based on mathematical logic. Specification topics covered in this course include: algebraic specifications and abstract model specifications, and initial specification and refinement towards implementation. The verification of software deals with using rigorous deduction in logic. This course covers many formal methods for program verification including, Hoare's method and Dijkstra's weakest preconditions. Finally, because different approaches and phases of development may require different techniques, the course also covers the application of formal methods to requirements analysis, testing, safety analysis, and object-oriented approaches.		
SWE 549	**Software Usability**	**3 (3+0)**
This course covers in-depth software usability including usability engineering concepts, usability requirements standards and user experience (UX) in the Software Engineering Process; User interface and/or experience design; User interface malfunctions analysis; Qualitative and quantitative methods for the evaluation of software system usability: Heuristic evaluation, cognitive walk through, formal usability experimentation and testing. Internationalization, and Accessibility. This course will also focus on the different software usability engineering aspects from the perspective of user interface design, user experience (UX) and analysis. The objective is a) how to develop software systems that are highly usable and improve user experience (UX); and b) How to do good research in software usability engineering. Students will be exposed to the research literature in the field of software usability.		
SWE 550	**Selected Topics in Software Engineering**	**3 (3+0)**
This course focuses on advanced and important software engineering topics that are not covered in the other courses. The content will be decided by the department each time the course is offered, in concordance with the faculty member teaching the course. Examples of topics than can be covered include patterns and antipatterns in software engineering, advanced re-engineering models and techniques, algorithms biasness and software fairness, data science & engineering for software engineers, etc. Students will be exposed to the research literature relevant to the topics covered.		

SWE 551	Human–Computer Interaction	3 (3+0)

Human Computer Interaction (HCI) integrates theories and methodologies from computer science, cognitive psychology, design, and many other areas. This course focuses on theories and methods related to the human aspects of design and use of computing systems. Course topics include: perceptual theory, motor theory, information processing theory, mental models, constructivism, distributed cognition, activity theory, small group psychology, embodied interaction, ethnography and ethnomethodology, and theories of experience. For each approach the course emphasizes scientific foundations, and applications in HCI. Case studies and research papers will be used to exemplify the methods presented and to give a context to the issues discussed. Students will be exposed to the research literature relevant to HCI and its applications in software engineering.

SWE 552	Real–time and Embedded Systems	3 (3+0)

This course is about creating software for embedded systems where real–time constraints are involved. It covers the unique aspects of embedded programming and includes interrupts, real–time control, synchronization, scheduling, resource handling, and fault–tolerance. The course explores also the tools used to develop and test embedded systems.

SWE 553	Enterprise System Architecture	3 (3+0)

This course addresses the required MSc SWE students' knowledge and skills necessary to design enterprise systems high–level and low–level architectures in alignment with the enterprise business architecture and processes. The concepts of strategic goal, business process and business planning constitute the basis of the course. These concepts are introduced through different Enterprise Architecture Frameworks (EAF) (e.g. Zachman, DODAF, FEAF, and TOGAF). SW Requirement Engineering (GORE, in particular), SW architecting and design, and IT deployment architecting are integrated with the different EAF to show where and how the SW engineer intervenes in the general process of implementing a new enterprise architecture or evolving an existing one. The course also explores the potential of the ERP commercial systems (SAP, Oracle, etc.), as well as a variety of relevant topics (e.g. UEML, Archimate, SOA governance, e–government, smart cities, and knowledge management).

SWE 555	Software Maintenance and Evolution	3 (3+0)

This course covers the important knowledge area of maintenance and evolution of software systems. The course covers fundamental concepts and techniques of maintenance and evolution including software maintenance taxonomy, Lehman's laws of evolution, software maintenance processes, change impact analysis, code smells and software refactoring, program understanding, Re–engineering concepts and techniques. Configuration Management will be also introduced in this course. Reading and presenting relevant research papers will be an important part of the course.

SWE 556	Web Engineering	3 (3+0)
Web applications are used daily by millions of people. This course is about how to develop these applications. It focuses on what distinguishes the development of web applications from the development of other types of applications. The development process will be presented along with a detailed description of the activities that compose it: requirements, user interaction, navigation, testing. Different issues important for web applications will be addressed: usability, reliability, scalability, and maintainability. The student will be exposed to the research literature relevant to these topics.		
SWE 557	**Reuse-based Software Engineering**	**3 (3+0)**
This course introduces the concepts and foundations of software reuse (dimensions of reuse, benefits, and drawbacks of reuse, application frameworks, software product lines, and application system reuse). It will also cover software components (component models, the composition of components, CBSE process, COTS and COTS product reuse, identification and selection of components), software variability as a factor affecting reusability of software (its scope, types, and issues), and Software as a Service (characteristics, advantages, and disadvantages). Free and open source software (their definition, differences, advantages, and disadvantages) will be also discussed. Finally, the course will cover measurements and metrics for reuse and CBS (e.g. reusability, amount of reuse, etc.) as well as reusability assessment models and a brief discussion on tool support for measurement.		
SWE 558	**Multimedia Software Systems**	**3 (3+0)**
This course will cover multimedia enabling technologies to understand, analyze, and build multimedia software systems by applying software engineering principles. It covers diverse topics including multimedia enabling technologies, services and applications; compression and networking technology in multimedia system, Multimedia and the Internet, Quality of Service (QoS) and Resource Management, Scheduling and synchronization, conferencing and collaboration tools, Multimedia Security, Multimedia content analysis, retrieval, and mining. Students will be exposed to the research trends in mobile software systems through reading and presenting research papers.		
SWE 561	**Service-Oriented Computing**	**3 (3+0)**
This course is concerned with concepts, theories, and techniques for Service-Oriented Computing (SOC). SOC is the paradigm for distributed computing that represents distributed software applications as a collection of services that can communicate with each other. The topics covered in this course include: standards related to web services and approaches for selection, composition, and management of web services, techniques for Service-Oriented Architecture (SOA), semantic web and ontologies, Quality of Service (QoS) issues, automated service composition, Peer-to-Peer (P2P) services and Cloud computing, and information security and privacy. Students will be exposed to the research trends in mobile software systems through reading and presenting research papers.		

SWE 562	Mobile Software Systems	3 (3+0)
This course presents the fundamental concepts to design and implement mobile software systems in wireless environment with user mobility. Main wireless networking concepts and protocols are presented in addition to the different computational models and distributed algorithms for mobile environment. Furthermore, the course introduces the different data delivery and dissemination techniques as well as the distributed file systems and data indexing techniques for mobile environment. Students will be exposed to the research trends in mobile software systems through reading and presenting research papers.		
SWE 563	**Dependable Software Systems**	**3 (3+0)**
Modern Software systems have extended in their distribution, mobility, and complexity. They are failure-prone and difficult to manage and thus hardly dependable. The dependability problems are hard to solve but must be dealt with regularly in order to detect, isolate and recover systems from these problems. This course covers the four dimensions of dependability: availability, reliability, safety, and security, and examines current research aiming at addressing challenges caused by software and hardware bugs and software misconfiguration. Students are expected to read and present/discuss recent research papers related to software dependability.		
SWE 564	**Software Data Mining**	**3 (3+0)**
This course focuses on mining and analyzing software data and software repositories related to various software artifacts such as use cases, code, bug reports, etc. It covers data mining concepts, algorithms, and applications on Software data. Data mining concepts include pattern recognition, anomaly detection, deep learning, and data prediction. It also covers widely used algorithms in data classification, association, and clustering. Students will learn different challenges related to mining Software data and how to apply data mining and analysis techniques on these data. The goal is to improve complex software systems development, management, and maintenance.		
SWE 565	**Emerging Technologies**	**3 (3+0)**
This course examines different emerging technologies and trends in the rapidly changing technology landscape that are relevant to software engineering. The content will be decided by the department each time the course is offered, in concordance with the faculty member teaching the course. Concepts and principles of various technologies such as cloud-based software systems, Internet of Things, Blockchain technology and Cryptocurrency, Quantum Computing, Evolutionary and nature-inspired optimization algorithms, etc. will be covered. The objective is to make students in software engineering embrace the new technological developments and trends that will impact software engineering in the future. Students will be guided towards the identification and characterization of core components, service structure, and workflow of these emerging technologies. Students will be exposed to the research literature relevant to the technologies covered.		