

Software Engineering Program

Course Title & No.	Brief Course Description
Foundation of Software Engineering 0103220 (3 Cr. H.)	This course introduces basic concepts of software engineering by focusing on software engineering process: development and maintenance. The course covers the software engineering lifecycle models and deliverables; requirements analysis and specification; architectural and detailed design; verification and validation and software maintenance issues. The Visio 2016 is used as CASE Tool for drawing diagrams covered in 'System Models' Chapter.
Software Requirements and Specifications 0103320 (3 Cr. H.)	This course introduces requirements engineering within software life-cycle: requirements elicitation and modeling issues and techniques; documentation and management of requirements; standards and CASE tools; cognitive and socio- organizational issues.
Software Design and Development 0103330 (3 Cr. H.)	The Software Design and Development course is designed to teach students the knowledge, understanding, skills and values to solve problems through the creation of software solutions. The course introduces students to the nature of software design, design process, agile-based design, architectural design and distributed architecture. It also introduces popular design frameworks, such as object-oriented design, function-oriented design, and aspect-oriented design. The course is concluded with user interface design and its contemporary design issues.
Formal Specifications and Design Methods 0103331 (3 Cr. H.)	Complex and large software system often have complicated control structures involving concurrency and real time interactions, which makes their development a challenging task. One of the best method of developing these kind of systems is using mathematical and logical tools (so-called formal methods). In this course, we will study formal methods in detail and their usage in requirements specification and software design. The course will start with introduction of formal specification, algebraic specification, and specification in Z language. Requirements and design strategies using Z language will be taught in detail. Design formal methods such as state charts and Petri-net models will also be covered thoroughly. The course will also introduce students with model checking using TLA+ and Alloy Analyzer. Object-Oriented formal language, OCL, will be taught with implementation

	on real-life complicated systems. The course will be concluded with software verification and validation methodologies.
Software Project Management 0103420 (3 Cr. H.)	This course develops the basic principles of project management, including concepts of project management process: initiation, planning, execution, monitoring & controlling, and closing. The course also introduces fundamentals of the main project management knowledge areas: integration, project scope, time management, cost estimation, quality, human resources and communications management. Moreover, this course presents project selection methods, importance of creating a project charter and scope management. The course also introduces Gantt chart and PERT for project planning and discusses different methods of cost estimation.
Object-Oriented Analysis & Design 0103430	This course introduces the object-oriented analysis and design, which includes the Unified Process development cycle; object oriented principles (encapsulation, inheritance, polymorphism), use case analysis; Unified Modeling Language (UML), dynamic and static diagrams; design principles (coupling and cohesion) and design patterns. The course includes a large-scale software-development project.
User Interface Design 0103431 (3 Cr. H.)	In this course, students will learn principles of designing, developing and testing appealing and effective user interface (UI) and user experience (UX) for desktop, web and mobile applications. They will learn about UI and UX design patterns, usability testing, and learnability. Issues and guidelines of designing modern and contemporary interfaces of small screened devices, such as Android and iPhone, will also be part of the course. The growing field of Big data demands effective data visualization on modern devices. We will learn methods and best practices to present such information in an effective manner on the applications along with creation and testing of modern Dashboards.

Agent-Based Modelling 0103432 (3 Cr. H.)	The course addresses the details of designing agent-based modelling (ABM) to solve specific problems of real systems, implementing models in Net Logo software, and analyzing models to develop theoretical understanding and predict system behaviors. The course starts with the basics of modeling by introducing a standard format—the Overview, Design concepts, and Details (ODD) protocol— for describing and thinking about designing ABMs. The course then introduces the basic concept of agent-based modeling and ways that the concept is often implemented. The course finishes by reinforcing and expanding Net Logo programming skills related to the concept.
Software Measurement and Testing 0103440 (3 Cr. H.)	This course is an introduction to software testing and metrics within the context of software quality engineering: module and unit testing; integration and acceptance testing; quality factors and metrics; verification and validation; review and inspections; reliability, security and safety assurance; and automated software testing.
Software Evolution and Maintenance 0103441 (3 Cr. H.)	This course provides concepts and advanced technologies in software evolution: Program comprehension; construction of reusable software; separation of concerns; techniques for reverse engineering and reengineering software; design for change layered design and incremental; and also analyze an existing system, explore possible change strategies, and construct a plan for evolving each of the systems major components.
Secure Software Engineering 0103443 (3 Cr. H.)	This course covers the design and implementation of secure software. Some of the topics covered are the characteristics of secure software, the role of security in the development lifecycle, designing secure software, and best security programming practices. Security for web and mobile applications will be covered.
SE Internship 0103480 (3 Cr. H.)	The Bachelor of Science in Software Engineering (BSSE) internship course is designed to let students apply their knowledge of mathematics, software engineering, and programming in real world professional situations. The students work as interns and practice team-based activities related to software engineering. The internship experience provides the students with an industrial training opportunity through cooperation between the university and the local industry, which can help them in choosing their career path based on hands-on industrial experience. The student should work on site for at least (480 hours (equivalent to 12 weeks of full-time work).
SE Capstone Project I 0103490 (1 Cr. H.)	Capstone-I starts with identification of a problem that could be resolved using software engineering techniques. Candidate solutions are considered to identify the most suitable solution depending on the nature, scope and context of the problem. The literature is reviewed to document the domain knowledge and existing similar solutions. It is followed by collection of requirements. Various techniques such as, surveys, interviews, questionnaires and study of relevant literature, are used to gather functional, non-functional and system requirements. The requirements are elaborated and specified using the selected software analysis and design methodology. This course concludes with preparation of an early design document and documentation of all the research work done during the semester. The team (2-4 members) is supervised by a faculty member. Several intermediate deliverables in addition to a final report and a presentation are submitted/delivered during the semester.
SE Capstone Project II 0103491 (2 Cr. H.)	Capstone-II starts with translation of specifications of the system identified in Capstone-I into implementable design. This involves designing system's requirements and modules in UML or another selected design framework. The design is approved by the supervisor and documented in the project's report. The next step is to develop prototypes of the system to get prospective users' feedback. The design is altered as per the feedback and is finalized. The implementation of the system starts in the next phase in the most suitable programming language and technology. The system is thoroughly tested and refactored after being implemented. The last step is to document all the activities of the project in form of a report. The same capstone 1 team (2-4 members) is supervised by the same capstone 1 supervisor. Several intermediate deliverables in addition to a final report and a presentation are submitted/delivered during the semester.