SOFTWARE ENGINEERING

Faculty of Computer Science				
Study programme:	**Computer Science**		Degree level:	**Engineer's degree full-time programme**
Specialization	**---**		Diploma path:	**2026/2027W - 2026/2027S**
Module name:	**Software Engineering** (Inżynieria oprogramowania)			
Module type:	**obligatory**	**Semester: 2**	ECTS:**5**	Module ID:**FCS-00014**
No. of hrs in semester:	Lecture (L) - **30** Classes(C) - **0** Specialization workshop (SW) - **30** Project (P) - **0** Laboratory classes (LC) - **0** Seminar (S) - **0**			
Prerequisites	Object Oriented Programming (FCS-00012),			

Aims and objectives:	The goal of the lecture is to acquaint students with the entire process of the creation and use of information systems. Students should understand that programming is only a component of this process, and that the success of the project affect all phases of the software life cycle. The lecture also includes a mini-course showing the use of Unified Modeling Language (UML) in modeling and designing systems. The purpose of the specialization workshop is to provide a practical introduction to modeling and design in UML using the CASE tool. In the first part of the course, UML diagrams are created based on given scenarios, while, in the second part, the acquired skills are verified by creating (preliminary) project of a selected information system.		
Forms of teching activities::	lecture, specialization workshop,	Assessment:	Evaluation must be relevant to the intended learning outcomes: Lecture: written exam (2 practical tasks - UML diagrams, and 3 theoretical questions) the condition to take an exam is passing practice laboratory; Practice laboratory: on the basis of short tests during classes and prepared in teams the project report.
Module content:	Lecture: objectives of software engineering (SE), reason of creating SE, methods and methodology, CASE tools; introduction to UML: use cases diagrams, activity diagrams, classes and objects diagrams, packages, interaction and state diagrams, physical diagrams: components and deployment; life cycle of the software (models: waterfall, spiral, COTS, ...); requirements engineering for information systems (methods of gathering information, the functional and non-functional requirements); modeling and design of systems, system implementation, testing, verification and validation of software (dynamic and static tests); ensuring software quality and software metrics; documentation, installation, deployment and maintenance of software; reliability of information systems, management of development projects, risk management in projects. Specialization workshop: sample CASE tools; tasks: use case diagram, describing use cases, class diagram, activity diagram, state diagram, interaction diagrams, physical diagrams (components and deployment); project: discussion on the topic of the group task, specifying the objectives and scope of the designed system and benefits of its implementation, creating and describing use case diagrams, designing the user interface, creating a class diagram, identifying attributes and methods, use cases realization, interaction diagrams, state diagrams, specifying non-functional requirements and technology suggestions, work plan, risk analysis, presenting the project.		
Teaching methods:	project method, programming, lecture problem, informative lecture,		

Learning outcomes

Symbol	Specify min. 4, max. 8 learning outcomes in the following order: knowledge – skills – competence. Each learning outcome must be verifiable	Reference to the programme learning outcomes of education
LO1	knows and understands the software engineering principles, methods and techniques used in the design of information systems. Knows the software life cycle model. Knows and understands the processes of its development, deployment and maintenance, and related methods of management and organization of work. Knows modelling languages and computer tools to support design.	INF1_W06 INF1_W09
LO2	knows and understands the processes and rules for the management of information technology projects. Knows the rules of the planning of the implementation system. Knows the techniques for estimating project costs and time needed for implementation of the mandated tasks.	H1_W02 INF1_W06 INF1_W09 INF1_W12
LO3	is able to design and plan implementation, testing and deployment of the information system and its components using appropriate methods, techniques and tools, taking into account the specified criteria and economic utility.	INF1_U04 INF1_U06 INF1_U07 INF1_U13 INF1_U15
LO4	is able to develop documentation of the project: requirements specification, architecture of the system, a description of the implementation and technology, user manual. Is able to work in a team and independently.	INF1_U04 INF1_U07 INF1_U13 INF1_U17

No. of learning outcome	Methods of assessing the learning outcome	Type of teaching activities (if more than one) during which the outcome is assessed
LO1	written exam	L
LO2	written exam	L
LO3	documentation of the project, discussion about the project, work during classes	Sw
LO4	documentation of the project, discussion about the project, work during classes	Sw
	1 - Attendance at lectures	30

Student's workload (in hours)	**2** - Attendance at laboratories		30
	3 - Elaboration of reports from the laboratories and execution of homeworks	None	49
	4 - Participation in student-teacher sessions	None	4
	5 - Performance of projects tasks (with presentation)	None	10
	6 - Preparation for the exam	None	2
		TOTAL:	**125**
Quantitative indicators	Student's workload - activities that require direct teacher participation: (4)+(1)+(2)	64	**ECTS** 2.6
	Student's workload connected with practical classes (5)+(3)+(2)	89	3.6

Basic references:	1. I. Sommerville, Software engineering, Pearson Education, 2004. 2. D. Pilone, N. Pitman, UML 2.0 in a Nutshell, O'Reilly, 2005, online: http://it-ebooks.info/book/154/. 3. Hans van Vliet, Software engineering: principles and practice, John Wiley & Sons, 2008. 4. Grady Booch, James Rumbaugh, Ivar Jacobson, The unified modeling language user guide, Addison-Wesley Publ., 1999. 5. J. Phillips, IT project management : on track from start to finish, McGraw Hill Professional, 2010.
Further reading	1. M. E. Bays, Software Release Methodology, Prentice Hall PTR, 1999. 2. P. Graessle, H. Baumann, P. Baumann, Uml 2.0 in Action: A Project-based Tutorial, Packt Publishing, 2004. 3. J. Schmuller, Sams Teach Yourself UML in 24 Hours, Sams, 2004. 4. M. Fowler, K. Beck, D. Roberts, E. Gamma, Refactoring: Improving the Design of Existing Code, Addison Wesley Longman, 2012. 5. Frank F. Tsui, Essentials of Software Engineering, Jones & Bartlett Publishers, 2014.

Unit:	Software Department	Lecturer/ instructor	
Date of issuing the programme:	17th Feb. 2022	Author of the programme:	dr inż. Krzysztof Jurczuk,prof. dr hab. inż. Marek Krętowski

L - lecture, C - classes, LC - laboratory classes, P-project, SW - specialization workshop, S - seminar