

Bialystok University of Technology									
Field of study	Computer Science							Degree level and programme type	Engineer's degree full-time programme
Specialization/ diploma path	---							Study profile	academic
Course name	Software Engineering							Course code	FCS-00014
								Course type	obligatory
Forms and number of hours of tuition	L	C	LC	P	SW	FW	S	Semester	3
	30				30			No. of ECTS credits	6
Entry requirements	Object Oriented Programming (FCS-00012),								
Course objectives	<p>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.</p> <p>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.</p>								
Course content	<p>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.</p> <p>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.</p>								
Teaching methods	informative lecture, lecture problem, programming, project method,								
Assessment method	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.								
Symbol of learning outcome	Learning outcomes							Reference to the learning outcomes for the field of study	
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.							K_W06	
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.							K_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.							K_U06 K_K03	
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.							K_U06 K_K03	
Symbol of learning outcome	Methods of assessing the learning outcomes							Type of tuition 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	
Student workload (in hours)							No. of hours		
Calculation	1 - Attendance at lectures -							30	
	2 - Attendance at laboratories -							30	
	3 - Elaboration of reports from the laboratories and execution of homeworks -							20	
	4 - Participation in student-teacher sessions -							5	
	5 - Performance of projects tasks (with presentation) -							50	
	6 - Preparation for the exam -							15	
TOTAL:							150		
Quantitative indicators							HOURS	No. of ECTS credits	
Student workload - activities that require direct teacher participation							65 (2)+(4)+(1)	2.6	
Student workload - practical activities							100 (5)+(3)+(2)	4.0	
Basic references	<ol style="list-style-type: none"> I. Sommerville, Software engineering, Pearson Education, 2004. D. Pilone, N. Pitman, UML 2.0 in a Nutshell, O'Reilly, 2005, online: http://it-ebooks.info/book/154/. Hans van Vliet, Software engineering: principles and practice, John Wiley & Sons, 2008. Grady Booch, James Rumbaugh, Ivar Jacobson, The unified modeling language user guide, Addison-Wesley Publ., 1999. J. Phillips, IT project management : on track from start to finish, McGraw Hill Professional, 2010. 								

Supplementary references	<ol style="list-style-type: none"> 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. 	
Organisational unit conducting the course	Software Department	Date of issuing the programme
Author of the programme	dr inż. Krzysztof Jurczuk, prof. dr hab. inż. Marek Krętowski	Feb. 17, 2022

L – lecture, C – classes, LC – laboratory classes, P – project, SW – specialization workshop, FW – field work, S – seminar