DESCRIPTION OF A STUDY COURSE – SYLLABUS

Title of a course	**Software Engineering**				
Head of course	**PhD Marin Kaluža, College Professor**				
Study programme	**Professional undergraduate study Telematics**				
Status of a course	Obligatory				
Year of study	2.	**Semester**	IV	**ECTS credits**	6
Teaching plan (L + E + S+ Pr)	3L+3E				

Goals of a course

Acquiring knowledge and competencies in process modelling used in the information system development and software development process. Acquiring knowledge and competences on the optimization procedures of the designed data model. Acquisition of competencies for designing and maintaining information system project documentation. Acquiring knowledge and competences in the field of application of software tools for information system design and development, and tools for rapid software development.

Conditions for enrolling course

Databases, Programming I, Algorithms and data structures

Learning outcomes on a level of a study programme which includes course

Outcome 4: Use computer principles and methods related to the architecture and organization of computers and computer networks.
Outcome 5: Use computer principles and methods related to programming languages, databases, and operating systems.
Outcome 6: Design and implement desktop, web and mobile computer applications and computer programs for microcomputers and microcontrollers, with or without a database.
Outcome 10: Analyse and implement an information system in the field of telematics.
Outcome 12: Design and develop solutions for components, circuits and software for application in computer networks and information systems, with the preparation of supporting project documentation.
Outcome 15: Participate in teamwork and independently present professional content in written and spoken form in Croatian and English.

Expected learning outcomes on a level of a course

1. Use methods to model processes and data, and model optimization procedures in the design of project documentation for IS.
2. Differentiate types and describe software properties.
3. Describe the software development process, distinguish software development models, and explain and compare the scope of application of CASE and RAD tools.
4. Create project documentation for information system development.
5. Sketch the structure and draw a user interface template.
6. Use rapid software development tools and create software for engineered IS.

Content of a course

Concept of software engineering. Software engineering methodology. Development stages of system programming. Approaches to the development of system programming. Development project of the system programming. Organization of the programming team. Program specifications. Structural analysis of the system. Object oriented system analysis. Tools in the development of system programming (CASE tools). Requirement Engineering. Creating and designing of system programming. Management of changes. Quality bases of system programming. Economy bases of system programming.

Teaching modes	☒ lectures ☐ auditory exercises ☒ seminars and workshops ☐ distance learning ☐ field classes	☒ individual assignments ☐ multimedia and network ☐ laboratory ☐ supervisor's work ☐ other _____

Comments	

Students' obligations

Grading, evaluation and monitoring of students' work continuously during lectures and exams

Grading is based upon evaluation of course's learning outcomes' adoption. Grading is performed continuously during lectures and/or during exam, in compliance with the provisions of Regulation on the assessment of students.

Continuous check-up:

Outcomes	Theoretical exam	Practical exam 1	Practical exam 2	Project	Threshold	Max
Outcome 1	**5%**			**10%**	**7,5%**	**15%**
Outcome 2	**5%**				**2,5%**	**5%**
Outcome 3	**10%**				**5%**	**10%**
Outcome 4				**10%**	**5%**	**10%**
Outcome 5				**10%**	**5%**	**10%**
Outcome 6		**20%**	**20%**	**10%**	**25%**	**50%**
Percentage of ECTS	**1,2**	**1,2**	**1,2**	**2,4**		
Total	**20%**	**20%**	**20%**	**40%**	**50 %**	**100 %**

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Exam term:

Outcomes	Theoretical part	Practical Part 1	Practical part 2	Max
Outcome 1	**5%**		**10%**	**15%**
Outcome 2	**5%**			**5%**
Outcome 3	**10%**			**10%**
Outcome 4			**10%**	**10%**
Outcome 5			**10%**	**10%**
Outcome 6		**40%**	**10%**	**50%**
Percentage of ECTS	**1,2**	**2,4**	**2,4**	
Total	**20%**	**40%**	**40%**	**100 %**

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Grading:

A student has passed the exam if he has acquired at least 50% of anticipated credits of a specific learning outcome.

If a student has passed learning outcomes of all courses, the accomplished credits (percentages) of all passed learning outcomes are being added, while the final grade is defined upon following table:

Range of credits (percentages)	Numerical grade	ECTS grade
90,00 – 100,00	Excellent (5)	A
75,00 – 89,99	Very good (4)	B
60,00 – 74,99	Good (3)	C
50,00 – 59,99	Sufficient (2)	D
0,00 – 49,99	Insufficient (1)	F

Obligatory literature

1. Materials used in lectures and tutorials in the course Software Engineering; available on Moodle.
Additional literature