1.	Course title			**Software architectures**
2.	Course code			SI-I-12
3.	Study program			**Master Studies in Computer Science and Engineering - Software engineering**
4.	Unit offering the course			**FCSE**
5.	Undergraduate/master/PhD			**Master**
6.	Year/semester 1(2)/summer/elective			7. ECTS: **6**
8.	Teacher(s)			assoc. prof. dr. Dejan Gjorgjevikj, assist. prof. dr. Ljupcho Antovski
9.	Course prerequisites			None
10.	Goals (competences): To introduce the students to the software architectures and teach them how to build robust, scalable and reliable software intensive systems in an effective way. Upon competition of the course the students are expected: to have a clear perception of the impact of abstraction, modelling, architecture and design patterns in developing a software product; to select the optimal architecture and to apply the most relevant methods and technologies in implementing the software solution regardless of the its complexity and volume; to understand to be able to precisely describe the concepts and the principles of software architectures; to be able to recognize the common architecture styles in existing software systems; to be able to develop a software architecture and global design starting from given requirement or by reverse engineering; to be able to generate appropriate number of alternative architectures for a given problem, to analyse them and to select among them; to be able to evaluate commercial software tools and software components regarding their architecture; to use object-oriented models and tools for efficiently performing their activities as software architects; to utilize patterns, styles and frameworks in creating a software architecture; to be able to systematically evaluate given software architecture; have a clear understanding of the relation among the software architecture and the other disciplines of software engineering.			
11.	Course content: Software architectures – definition and overview, Quality of software and software architectures, Designing software architectures, Creating and analysing architectures, Various views in software architecture, Architecture description languages, Documenting software architectures, Revising software architectures, Architectural styles, Software architecture design patterns, Software architectural frameworks, Reusing software architectures.			
12.	Teaching methods: Lectures supported by slide presentations, interactive lectures, trainings (using lab equipment and software packages), team work, case studies, invited guests and lectures, individual practical assignments presentations, seminar paper, e-learning (forums, consultations).			
13.	Total available time			6 ECTS x 30 hours = 180 hours
14.	Distribution of the available time			60 + 0 + 120 = 180 hours
15.	Teaching activities	15.1.	Lectures	60 hours

		15.2.	Training (labs, problem solving), seminar and team work	0 hours
16.	Other activities	16.1.	Project work	45 hours
		16.2.	Self study	45 hours
		16.3.	Home work	30 hours

17.	Grading		
	17.1.	Tests	45 points
	17.2.	Seminar work/project (written or oral presentation)	45 points
	17.3.	Active participation	10 points

18.	Grading criteria	to 59 points	5 (five) (F)
		from 60 to 68 points	6 (six) (E)
		from 69 to 76 points	7 (seven) (D)
		from 77 to 84 points	8 (eight) (C)
		from 85 to 92 points	9 (nine) (B)
		from 93 to 100 points	10 (ten) (A)

19.	Final exam prerequisites	Successfully completed activities 15.1 and 15.2
20.	Course language	Macedonian and English
21.	Quality assurance methods	Internal evaluation and student questionnaires

22.	Literature					
	22.1.	Compulsory				
		No.	Authors	Title	Publisher	Year
		1.	Len Bass, Paul Clements, Rick Kazman	Software Architecture in Practice	Addison Wesley Longman	1998
		2.	Luke Hohmann	Beyond Software Architecture: Creating and Sustaining Winning Solutions	Addison Wesley	2003
		3.	Stephen T. Albin	The Art of Software Architecture: Design Methods and Techniques	John Wiley & Sons	2003
	22.2.	Additional				
		No.	Authors	Title	Publisher	Year
		1.				
		2.				
		3.				