

1.	Course title	Software design patterns		
2.	Course code	SI-Z-02		
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)/winter/compulsory	7. ECTS: 6		
8.	Teacher(s)	prof. dr. Suzana Loshkovska, assoc. prof. dr. Dejan Gjorgjevikj		
9.	Course prerequisites	None		
10.	Goals (competences): To introduce the students to the patterns for software design, the standard solutions for standard problems when designing software. To introduce the mechanisms for software evolution, refactorization and the implementation patterns to the students. Upon completion of the course the students will be able to understand, acknowledge the need and apply the cohesion and coupling techniques to the components of the object-oriented design; to describe and to successfully use the most common programming patterns; to categorize the design patterns according to their structure, intent, responsibility, construction and applicability; to design and construct large framework computer systems applying the meta-system approach of object-oriented programming; to design and implement an application selecting and applying the appropriate programming patterns.			
11.	Course content: Software design, Design patterns – definition and history, Observer Pattern, Template Method Pattern, Factory Patterns: Factory Method and Abstract Factory, Singleton Pattern, Facade Pattern, Visitor Pattern, Functors and the Command Pattern, Iterator Pattern, Composite Pattern, State and Strategy Patterns, Adapter Pattern, Proxy Pattern, Decorator Pattern, Chain of Responsibility Pattern, Concurrency Patterns, AntiPatterns, Implementation patterns, Role-based design, Composite design patterns, Design patterns in formal methods.			
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					
	Compulsory					
		No.	Authors	Title	Publisher	Year
	22.1.	1.	Gamma, et. al.	Design Patterns - Elements Of Reusable Object-Oriented Software	Addison- Wesley	1995
		2.	Steve McConnell	Code Complete: A Practical Handbook of Software Construction, 2 nd edition	Microsoft Press	2004
		3.	Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra	Head First Design Patterns	O'Reilly Media, Inc.	2004
	Additional					
		No.	Authors	Title	Publisher	Year
	22.2.	1.				
		2.				
3.						