1.	Course title			**Software Reliability**
2.	Course code			SOCD-I-04
3.	Study program			**System on Chip Design**
4.	Unit offering the course			**FCSE**
5.	Undergraduate/master/PhD			**Master**
6.	Year/semester 1(2)/winter/elective			7. ECTS: **6**
8.	Teacher(s)			Assist. Prof. Sonja Filiposka, Assist. Prof. Anastas Mishev, Assist. Prof. Ivan Chorbev
9.	Course prerequisites			None
10.	Goals (competences): After successfully completing the course, the student is expected to understand the basic concepts of error tolerance and the basic techniques employed for achieving error tolerant software or communication system. The student will have the skills to model and evaluate the error tolerance and security of different architectures as well as be able to locate the error sources and prevent error propagation while designing and developing reliable software.			
11.	Course content: Software reliability and availability. Using error tolerant software. Software development cycle. Software errors models, error sources in the software. Error-free software development strategy. Using coding standards. Different software structures, module decomposition, partitioning, module closure, atomic actions. Error detection, timing checks, reversal checks. Error detection and correction codes. Logic checks, structural checks. Exception handling. Exception handling system design. Debugging using induction and deduction. Information logging. Using logging with debugging and for error location purposes. Testing, using auto testing. Unit tests, integration testing, acceptance testing. One-version reliable software design techniques. Check-in and restart, process pairs, input data diversity. N-version programming. Recovery blocks, self-check programming, consensus, t/(n-1)-variant programming.			
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			30 + 15 + 135 = 180 hours
15.	Teaching activities	15.1.	Lectures	30 hours
		15.2.	Training (labs, problem solving), seminar and team work	15 hours
16.	Other activities	16.1.	Project work	60 hours
		16.2.	Self study	25 hours
		16.3.	Home work	50 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.	Israel Koren, C. Mani Krishna	Fault-Tolerant Systems	Morgan Kaufmann	2007
		2.	Laura L. Pullum	Software Fault Tolerance Techniques and Implementation	Artech House Publishers	2001
		3.	Kopetz, Hermann	Real-Time Systems, Design Principles for Distributed Embedded Applications	Springer	2011
	22.2.	Additional				
		No.	Authors	Title	Publisher	Year
		1.	Mitch Lacey	The Scrum Field Guide: Practical Advice for Your First Year (Agile Software Development Series)	Addison-Wesley Professional	2012
		2.	Dan Skwire	First Fault Software Problem Solving: A Guide for Engineers, Managers and Users	Opentask	2009
		3.	Sam Guckenheimer, Neno Loje	Agile Software Engineering with Visual Studio: From Concept to Continuous Feedback (2nd Edition) (Microsoft .NET Development Series)	Addison-Wesley Professional; 2 edition	2011