

Study programme(s): Information Technologies				
Level: Master				
Course title: Software Evolution				
Lecturer: Miloš M. Radovanović				
Status: elective				
ECTS: 7				
Requirements: none				
Learning objectives The goal of this course is to present and critically analyze current techniques for software evolution and provide students with practical experience in using a set of tools known as FermaT.				
Learning outcomes <i>Minimum:</i> At the end of the course it is expected from a successful student to be capable of critically evaluating the current basics of software evolution, adopt reengineering techniques for software migration and abstraction, and develop an integrated approach for software evolution life cycles. <i>Desirable:</i> At the end of the course it is expected from a successful student to demonstrate the ability to apply transformation rules in order to migrate a temporally and economically critical system, and acquire practical experience in the use of an industrial-strength tool such as FermaT.				
Syllabus <i>Theoretical instruction</i> Theoretical basis and classifications of software evolution, evolution within software development life cycles, Lehman's laws of evolution, software comprehension techniques, abstraction, slicing, refactoring, Wide Spectrum Language (WSL) and software transformation, transformation theory and its implementation, tools, and migration of software. <i>Practical instruction</i> Acquaintance with tools such as FermaT and analysis of study examples.				
Literature 1. H. Yang, M. Ward. <i>Successful Evolution of Software Systems</i> . Artech House, 2003 2. M. Fowler. <i>Refactoring: Improving the Design of Existing Programs</i> . Addison-Wesley, 1999				
Weekly teaching load				
Lectures: 2	Exercises: 1	Practical Exercises: 2	Student research: 0	Other: 0
Teaching methodology Lectures are held using classical presentation methods involving a projector. During exercises classical teaching methods involving a projector are used to analyze study examples. Also, the principles of application of studied topics are practiced on the computer, through acquaintance with the use of recommended tools. Students complement their knowledge through research into selected topics, and are tested through a written test, solution of practical problems, and preparation of a seminar paper that is defended at the end of the course.				
Grading method (maximal number of points 100)				
Pre-exam obligations	points	Final exam	points	
Test	20	Seminar paper	60	
Practical problems	20			