

Faculty of Science and Mathematics / COMPUTING AND INFORMATION TECHNOLOGY / COMPILERS

Course:	COMPILERS			
Course ID	Course status	Semester	ECTS credits	Lessons (Lessons+Exercises+Laboratory)
1348	Mandatory	5	6	3+2+0
Programs	COMPUTING AND INFORMATION TECHNOLOGY			
Prerequisites	None.			
Aims	This course covers the design and implementation of translator-oriented systems software, focusing specifically on compilers, with some time spent on related topics such as interpreters and linkers.			
Learning outcomes	At the end of the course, the participant is expected to be able to: 1. Describe the design of a compiler/interpreter including its phases and components [Familiarity] 2. Use regular expressions and context-free grammars to specify the syntax of languages [Usage] 3. Identify the similarities and differences among various parsing techniques, grammar transformation techniques and type checking methods [Familiarity] 4. Distinguish between methods for scope and binding resolution and parameter passing [Familiarity] 5. Explain how programming language implementations typically organize memory [Familiarity] 6. Design and implement interpreter/compiler for simple language using declarative tools to generate parsers and scanners. [Usage]			
Lecturer / Teaching assistant	Goran Šuković, Savo Tomović.			
Methodology	The course lasts 14 weeks and consists of two 45-minutes session per week of face-to-face lectures together with a two 45-minute recitation class.			
Plan and program of work				
Preparing week	Preparation and registration of the semester			
I week lectures	Introduction. Compilers and interpreters.			
I week exercises	MIPS intro.			
II week lectures	Grammars and languages.			
II week exercises	MIPS: function call, recursion.			
III week lectures	Lexical Analysis			
III week exercises	Regular Expressions. DFA. NFA.			
IV week lectures	Syntax Analysis – "top-down" parsers.			
IV week exercises	RE to NFA conversion. NFA to DFA conversion. DFA optimization.			
V week lectures	Syntax Analysis – "Bottom-up" parsers. LR(0), SLR(1).			
V week exercises	Intro to Flex/Lex.			
VI week lectures	Syntax Analysis – LR(1), LALR.			
VI week exercises	Flex examples.			
VII week lectures	Midterm.			
VII week exercises	Midterm.			
VIII week lectures	Semantic Analysis.			
VIII week exercises	Bison/Yacc examples.			
IX week lectures	Type checking.			
IX week exercises	Symbol table.			
X week lectures	Runtime environment.			
X week exercises	Type checking using Bison/YACC.			
XI week lectures	TAC			
XI week exercises	TAC examples.			
XII week lectures	Code generation			
XII week exercises	Code generation examples.			

XIII week lectures	Code generation (cont.)					
XIII week exercises	Code generation with Bison/YACC					
XIV week lectures	Intro to dataflow analysis. Loop optimization.					
XIV week exercises	Optimization – examples.					
XV week lectures						
XV week exercises						
Student workload	Weekly: 5x40/30 = 6 hours 40 minutes, Lectures: 1 hour 30 minutes, Labs: 1 hour 30 minutes, Other: 0, Individual works: 3 hours 40 minutes.					
Per week			Per semester			
6 credits x 40/30=8 hours and 0 minuts 3 sat(a) theoretical classes 0 sat(a) practical classes 2 excercises 3 hour(s) i 0 minuts of independent work, including consultations			Classes and final exam: 8 hour(s) i 0 minuts x 16 =128 hour(s) i 0 minuts Necessary preparation before the beginning of the semester (administration, registration, certification): 8 hour(s) i 0 minuts x 2 =16 hour(s) i 0 minuts Total workload for the subject: 6 x 30=180 hour(s) Additional work for exam preparation in the preparing exam period, including taking the remedial exam from 0 to 30 hours (remaining time from the first two items to the total load for the item) 36 hour(s) i 0 minuts Workload structure: 128 hour(s) i 0 minuts (cources), 16 hour(s) i 0 minuts (preparation), 36 hour(s) i 0 minuts (additional work)			
Student obligations						
Consultations			Room 128.			
Literature			Torczon, Cooper – Engineering a Compiler, 2nd Edition (Morgan Kaufmann, 2011) Appel – Modern Compiler Implementation in Java (2nd edition), Cambridge University Press, 2002. Aho, Sethi, Ullman – Compilers: Principles, Techniques and Tools, 2nd Edition (Pr			
Examination methods			- 6 Homewoks (3-5% each, programming and pen-and-pencil) = 20% - Midterm 40% - Final exam 40%			
Special remarks			The lecturer is able to offer course in English and Russian.			
Comment			www.pmf.ac.me, prevodioci@rc.pmf.ac.me			
Grade:	F	E	D	C	B	A
Number of points	less than 50 points	greater than or equal to 50 points and less than 60 points	greater than or equal to 60 points and less than 70 points	greater than or equal to 70 points and less than 80 points	greater than or equal to 80 points and less than 90 points	greater than or equal to 90 points