

Study programme(s): Computer Science				
Level: Bachelor				
Course title: Basics of Languages and Translation				
Lecturer: Mirjana Ivanović				
Status: Elective				
ECTS: 4 ECTS				
Requirements: Previous 4 semesters completed.				
Learning objectives The objective of the course is to introduce students to basic concepts of computer languages, algorithms, and techniques necessary for translation of different kinds of computer languages.				
Learning outcomes <i>Minimum:</i> Successful students should be capable to understand basic concepts of computer languages and their translation. <i>Desirable:</i> At the end of the course it is expected that successful students deeply understand and are able to implement some essential algorithms related to computer language translation.				
Syllabus <i>Theoretical instruction</i> Systematization and classification of computer languages and their concepts. Core issues in language design. Formal grammars and their role in computer languages translation. Deterministic finite automata and push-down automata and their role in computer language translation. Fundamental concepts and elements of computer language translation (Specifying syntax; Scanning; Parsing; Names, Scopes, and Bindings including symbol tables and garbage collection; The role of the semantic analyzer; Target machines). Techniques and methods for intermediate code representation. Domain specific languages and their translation. Different tools for language translation. <i>Practical instruction</i> During exercises numerous illustrative examples, that follow topics presented during theoretical classes, will be presented to the students.				
Literature <i>Recommended</i> 1. Terence Parr, Language Implementation Patterns: Create your own domain-specific and general programming language, The Pragmatic Programmers, 2010. 2. Martin Fowler, Domain-specific languages, Addison-Wesley Signature Series, 2010. 3. Michael L. Scott, Programming language pragmatics, Morgan Kaufmann, 2009.				
Weekly teaching load 3				
Lectures: 1	Exercises: 2	Practical Exercises: /	Student research: /	Other: /
Teaching methodology Theoretical classes are based on the classical teaching model involving a projector and .ppt presentations. At exercises solutions of practical programming problems involving algorithms and techniques covered by theoretical classes will be illustrated. To approach the oral exam students have to pass pre-exam obligations consisting of a home work i.e. practical programming tasks using particular tool for language translation. Several mini tests will be conducted during classes. At the oral exam students are expected to demonstrate the understanding of the topics covered by the course.				
Grading method (maximal number of points 100)				
Pre-exam obligations	points	Final exam	points	
Mini tests and practical task	40	Oral examination	60	