

Study programme(s): Information Technologies				
Level: Bachelor				
Course title: Advanced Functional Programming				
Lecturer: Zoran D. Budimac				
Status: elective				
ECTS: 6				
Requirements: None				
Learning objectives To introduce students to advanced programming techniques in functional and hybrid programming languages such as Haskell, Erlang, Scala or domain specific embedded languages. The course will have two spotlights: a) theoretical and b) practical with emphasis to large scale programming.				
Learning outcomes <i>Minimal:</i> at the end of the course it is expected that successful student is able to understand advanced concepts of functional programming languages and advantages of hybrid programming languages. <i>Optimal:</i> at the end of the course, apart from minimal outcomes, it is expected that successful student is able to understand advantages and flaws of practical usage of functional programming languages in a large scale programming.				
Contents of the course <i>Theoretical instruction</i> Introduction to advanced language construction of functional and hybrid programming languages and possible ways how two paradigms function in one programming language. Overview of some characteristic programming languages. Monads, functors, automatic transformations, parallelization, verification, types. Advantages of using such languages in large scale programming efforts. <i>Practical instruction</i> Work on a large scale example written in one of the characteristic languages. Analysis and adding new functionality.				
Literature 1. O'Sullivan, B., Stewart, D., Goerzen, J., Real World Haskell, O'Reilly, 2008. 2. Martin Odersky, Lex Spoon, and Bill Venners, Programming in Scala, Addison-Wesley, 2016. 3. Cesarini, F., Thompson, S., Erlang Programming, O'Reilly, 2009.				
Classes of active instruction				
Lectures: 2	Exercises: 0	Practical exercises: 2	Self study research work: 0	Other classes: 0
Teaching methods During the theoretical instruction, classical methods are used with the usage of projector. Practical instruction is organized around analysis of a practical example, which is analyzed and then extended its functionality.				
Final mark (max. 100)				
Pre-exam obligations	points	Final exam	Points	
Practical assignments	60	Oral exam	40	