

Study programme(s): Computer science			
Level: academic bachelor studies			
Course title: Programming languages and paradigms			
Lecturer: Zoran D. Budimac			
Status: obligatory			
ECTS: 7			
Requirements: None			
Learning objectives Introduction of historical and practical reasons that lead to emergence of many different programming languages, clarification of specifics, similarities and differences of several programming paradigms (object-oriented, functional, logic, multi-paradigm), with detailed description of most important characteristics of their most influential representatives.			
Learning outcomes <i>Minimal:</i> At the end of the course it is expected that successful student will be able to understand fundamental concepts of various programming languages and understands importance of different programming paradigms. <i>Desirable:</i> At the end of a course it is expected that a successful student will be able to understand concepts of programming languages, to understand concepts of programming languages and importance of different programming styles, as well as to demonstrate skills of specific program development in several programming paradigms.			
Syllabus <i>Theoretical instruction:</i> History of programming language development. Procedural and non-procedural programming languages. Characteristics of programming languages and most important differences between them. Detailed and comparative overview of several programming styles (functional, logic, multi-paradigm,...) and their typical representatives. Syntax and semantics. Basic notions and mathematical background. Data structures. <i>Exercises</i> Comparative overview of different approaches to implementation of standard programming tasks, and illustration of different philosophies behind every mentioned paradigm using typical examples. Testing of given programs, tools, discussion of applicability etc. Individual practical exercises: data types, statements, data structures.			
Literature <i>Recommended</i> 1. Krishnamurti, S., Programming languages: application and interpretation, Brown university, 2007. 2. Gabrielli, M., Martini, S., Programming Languages: Principles and Paradigms, Springer, 2010.			
Weekly teaching load			
Lectures: 2	Exercises: 1	Practical Exercises: 2	Student research: Other:
Teaching methodology Classical teaching methods are used for theoretical instruction using video beam projector. Theoretical exercises are used to discuss and analyze principles, typical problems and their solutions and model own programs. Practical exercises are used for individual application of taught techniques.			
Grading method (maximal number of points 100)			
Pre-exam obligations	Points	Final exam	points
Assignments	30	Oral exam	40
Tests	30		