

Study programme(s): Teaching Informatics (IC)				
Level: master				
Course title: Teaching methods in programming				
Lecturer: Mirjana K. Ivanović, Miloš M. Radovanović				
Status: obligatory				
ECTS: 7				
Requirements: None				
Learning objectives Enabling students to understand the basic teaching principles in programming and different programming techniques.				
Learning outcomes <i>Minimal:</i> Students should be able to demonstrate understanding of the basic concepts of (computer) programming, ability to analyze and define problems, and create solutions and present them using the appropriate teaching methods. <i>Optimal:</i> Students should be able to demonstrate understanding of the basic concepts of (computer) programming, ability to analyze and define real problems based on logic, and to create efficient, elegant solutions, on a very high level, using the appropriate teaching methods.				
Syllabus <i>Theoretical instruction</i> Forms of thinking in the programming process. Programming styles and programming languages that support them. Syntactic differences. The focus of programming languages on solving specific problems. Object-oriented programming. Overview of object oriented programming languages that are used in schools. Topics to be studied in schools (class, object, encapsulation, inheritance, polymorphism). Suitable tasks to illustrate the concepts introduced. Functional programming and overview of the functional programming languages. Basic terms in functional programming (lambda calculus, first-class functions, higher-order functions). Logic programming and basic terms in logic programming (predicate calculus, data representation). Comparative analysis of solutions using different programming styles. Environments and tools for program execution visualization. Development environment and other tools to support program development. <i>Practical instruction</i> Presentation of the basic concepts of programming languages, creating different software solutions and their comparative analysis. Comparative analysis of complex programmes in several programming paradigms.				
Literature Steve McConnell: <i>Code Complete</i> , Microsoft Press, A Division of Microsoft Corporation, One Microsoft Way, Redmond, Washington, 1993. Velimir Sotirović: <i>Metodika Informatike</i> , Faculty of Engineering "Mihajlo Pupin", Zrenjanin, 2000. Proceedings of the conference on history of programming languages. Ravi Sethi: <i>Programming Languages: Concepts and Constructs</i> , Addison Wesley, 1998. M. Stanković: <i>Programming languages</i> , Faculty of Electronic Engineering, Niš, 2000.				
Weekly teaching load				Other:
Lectures: 2	Exercises: 4	Other forms of teaching:	Student research:	
Teaching methodology Lectures are organized using classical teaching methods with the use of a projector. Most important principles of programming are explained and illustrated through appropriate examples.				

At the exercises, outlined principles are practiced; illustrative examples are analyzed, and own solutions are modelled. Solutions created using different programming styles are analyzed. During the exercises, students apply learned techniques by creating different applications, whose complexity and applicability increases during the semester. Students analyze and use a variety of tools for programme results visualization. Student knowledge is tested in two colloquia and several practical tasks. At the oral exam, students show understanding of the basic programming principles in different programming styles.

Grading (maximum number of points 100)

Pre-exam obligations	points	Final exam	points
Colloquia	30	Oral exam	40
Seminar(s)	30		