Course title: Teaching methods in programming

Lecturers: Mirjana K. Ivanović, Zoran D. Budimac

Status: elective

ECTS: 15

Requirements:

Learning objectives

Enabling the student to understand basic teaching principles in programming and different programming techniques

Learning outcomes

Minimal: At the end of the course it is expected from a successful student to demonstrate understanding of the basic concepts of (computer) programming, ability to analyze and define problems and create implementation of solutions and present them using appropriate teaching methods.

Desirable: At the end of the course it is expected from a successful student to demonstrate understanding of the basic concepts of (computer) programming, ability to analyze and define real problems based on logic, and to create efficient and elegant solutions on a very high level using appropriate teaching methods.

Syllabus

Theoretical instruction - 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 analyzes of solutions using different programming styles. Environments and tools for program execution visualization. Development environment and other tools to support program development.

Student research

- Presentation of the basic concepts of programming languages, creating different software solutions and their comparative analysis. Comparative analysis of complex programs in several programming paradigms.

Literature

Suggested:

1. Guide to Teaching Computer Science, O.Hazzan, T.Lapidot, N.Ragonis, Springer Verlag London, 2011.
2. Stratosphere: Integrating Technology, Pedagogy, and Change Knowledge, M.Fullan, ISBN-10: 0132483149, 2012.
3. Transforming Education with New Media: Participatory Pedagogy, Interactive Learning, and Web 2.0, P. DePietro, ISBN-10: 1433117940, 2013.
4. Articles from international and domestic journals, selected book chapters, specifically prepared texts for this purposes, and materials from international and domestic conferences.

Weekly teaching load	**Lectures**: 5	**Student research**: 5

Teaching methodology

During lectures, classical methods of teaching that includes video-beams are used. Key concepts of programming are described and illustrated through appropriate examples. During theoretical exercises, presented principles are exercised, illustrative examples are analyzed, and personal solutions are modelled. Solutions realized in different programming paradigms are analyzed. During practical exercises, students apply presented techniques developing various applications where its complexity and possibilities grow during the semester. Students analyze and use various tools for visualization of the results of applied programs. Students knowledge is tested by two colloquias, while at practical exercises

<table>
<tr><td colspan="4">students solve practical problems, which is also valued. At oral exam, students should present complete understanding of basic principles of various programming styles.</td></tr>
<tr><td colspan="4" align="center"><b>Grading (maximal number of points 100)</b></td></tr>
<tr><td><b>Pre-exam requirements</b></td><td>points</td><td><b>Final exam</b></td><td>points</td></tr>
<tr><td>Seminar papers</td><td><b>30</b></td><td>Oral exam</td><td>40</td></tr>
<tr><td>Colloquia</td><td><b>30</b></td><td></td><td></td></tr>
</table>