<table>
<tr><td>NAME OF THE COURSE</td><td colspan="5">Methods of Teaching Informatics II</td></tr>
<tr><td>Code</td><td>PMIK60</td><td>Year of study</td><td colspan="3">GU-2</td></tr>
<tr><td>Course teacher</td><td>izv. prof.dr. sc. Ivica Boljat</td><td>Credits (ECTS)</td><td colspan="3">6,0</td></tr>
<tr><td rowspan="2">Associate teachers</td><td rowspan="2">Monika Mladenović</td><td rowspan="2">Type of instruction (number of hours)</td><td>L</td><td>S</td><td>E</td><td>F</td></tr>
<tr><td>30</td><td>30</td><td>30</td><td></td></tr>
<tr><td>Status of the course</td><td></td><td>Percentage of application of e-learning</td><td colspan="3">20</td></tr>
<tr><td colspan="6">COURSE DESCRIPTION</td></tr>
<tr><td>Course objectives</td><td colspan="5">Theoretically and practically train students for quality preparation, realization and analysis of the teaching process based on the results of scientific research in the field of information education</td></tr>
<tr><td>Course enrolment requirements and entry competences required for the course</td><td colspan="5">It is desirable to know the Methods of teaching computer science</td></tr>
<tr><td>Learning outcomes expected at the level of the course (4 to 10 learning outcomes)</td><td colspan="5">Students will be able:<br>- Apply modern tools and robots to teach programming<br>- Apply problem-solving techniques and role variables in algorithms<br>- Identify potential misunderstandings and compile a test for their detection<br>- Learn the techniques of efficient code tracking<br>- Identify difficulties in understanding recursion and apply appropriate teaching models<br>- The advantages and disadvantages of teaching OOP before procedural and first teaching the methodology, only the language<br>- Use visualization of teaching and self-learning algorithms based on the theory of multimedia learning<br>- Select and operationalize variables in SPSS or PSPP, enter data, choose the appropriate method, get results, and interpret them correctly.</td></tr>
<tr><td>Course content broken down in detail by weekly class schedule (syllabus)</td><td colspan="5">1. Application of statistics in the study of computer science - sample, types of errors and dependence on sample size, type of scales, Kolmogorov - Smirnov test, square, Mann-Whitney, Kruskal-Wallis, t-test, variance analysis, factor hierarchical cluster analysis , Pearson's and Spearman's correlation, regression analysis (2 + 2 + 4)<br>2. Basic Literature and Classification of Research - Valentine, Fincher-Petre. Qualitative Methods (emphasis on phenomenography) in CSER. (2 + 2 + 0)<br>3. Teaching programming - (tools, visualizations, robots, and their didactic background). ALICE, JKarelRobot. Taxonomy of Programming Languages and Learning Programming Environments. Comparison of PJ criteria and results. Robot's effectiveness in teaching programming - examples of research. Legomindstorm NXT-G. - seagway, smoothfollow). Snap, Enchanting, RoboMind. Alice3 and Intermediate Transfer). Problem solving techniques. Role of variables. (5 + 6 + 8)<br>4. Typical initial errors in programming - classifications and causes. Comparison with experts. Misconception about the efficiency and correction program. Researching programming skills and tracing. (3 + 4 + 4)<br>5. Recursion - basic cases - typical errors, student difficulties and their causes. Why is it difficult - examples. Mental models of recursion. Models of teaching recursion - model of small people and nested frames. (4 + 4 + 4)</td></tr>
</table>

| | |
|---|---|
| | 6. Teaching Object Programming - Implementation Steps (Classes, Objects, Inheritance, Encapsulation, Reuseability), why prior to methodology rather than language, which is not recommended to use, which approach to use before (object or procedural). Phenomenogic study - understanding the concept of class, object. (2 + 2 + 0)<br>7. Visualization - application in teaching algorithms and data structures. - Examples (binary tree width and depth, fast and merge, heap-sort, AVL, Dijkstra-algorithm, hashing, Huffman's code). Research of real effects of visualization: meta-analysis. Levels of student involvement. HalVis - structure, characteristics. Theory of multimedia learning - principles. Cognitive Overload - Causes, Reduction Methods. (3 + 3 + 4)<br>8. Teaching Computer Networks - Which Key Concepts, Which Approaches Use, Recommendations of Phenomenological Study for Good Teaching. (2 + 2 + 0)<br>9. LOGISIM - principle of work, transition from the logical level to the table of truths and logical expression - example of a 4-bit comparator) (0 + 0 + 4)<br>10. Cognitive Models. Cognitive processes of students while programming. Spatial Intelligence (mapping) and programming success. Visualization in teaching the computer architecture. (2 + 2 + 0)<br>11. Women in CS. Reasons for subdivision. Attitudes of women about computer science. (1 + 1 + 0)<br>12. Abstraction - why is it important (2 + 2 + 0)<br>13. How to integrate CSE scientific research results into a curriculum. Curriculum goals, choice of topics and pedagogical strategies, teacher preparation, curriculum design principles, successful mass implementation strategies, examples of some countries, CS prejudice, Frankov Framework for Critical Analysis of Educational Policies and Reforms .. (2 + 0 + 0) |
| Format of instruction | ☐ lectures<br>☒ seminars and workshops<br>☒ exercises<br>☐ on line in entirety<br>☐ partial e-learning<br>☐ field work<br>☐ independent assignments<br>☐ multimedia<br>☐ laboratory<br>☒ work with mentor<br>☒ homework assignments |
| Student responsibilities | Attending all forms of teaching, teaching practice, oral exam. |

Screening student work *(name the proportion of ECTS credits for each activity so that the total number of ECTS credits is equal to the ECTS value of the course)*

| Name | Ects | Name | Ects | Name | Ects |
|---|---|---|---|---|---|
| Class attendance | 2 | Research | | Experimental work | |
| Oral exam | 3 | Report | | Homework assignments | |
| Seminar essay | | Essay | | | |
| Tests | | Practical training | 1 | | |
| Written exam | | Project | | | |

| | |
|---|---|
| Grading and evaluating student work in class and at the final exam | Oral exam (75%), teaching practice (25%). |

<table>
<tr><td rowspan="4">Required literature (available in the library and via other media)</td><td>Title</td><td>Number of copies in the library</td><td>Availability via other media</td></tr>
<tr><td>1. Boljat, I., Metodika nastave matematike II- predavanja, 2014.</td><td>0</td><td>yes</td></tr>
<tr><td>2. Hazzan, O., Lapidot, T., Ragonis, N., Guide to teaching computer science: an activity-based approach, Springer, 2011.</td><td>1</td><td>yes</td></tr>
<tr><td>3. Fincher, S., Petre, M., Computer science education research, Taylor & Francis 2004.</td><td>1</td><td></td></tr>
<tr><td>Optional literature (at the time of submission of study programme proposal)</td><td colspan="3">Journals. Computers & Education, ACM Transactions on Computing Education The Computer Science Education Journal<br>Confesrences: SIGCSE (Special Interest Group on Computer Science Education) ITiCSE (Innovation and Technology in Computer Science) ISSEP (Informatics in Secondary Schools: Evolution and Perspective)</td></tr>
<tr><td>Quality assurance methods that ensure the acquisition of exit competences</td><td colspan="3">Conversation with students, student evaluation using anonymous poll, student success on the exam, self-assessment.</td></tr>
<tr><td>Other (as the proposer wishes to add)</td><td colspan="3"></td></tr>
</table>