NAME OF THE COURSE	**Programming II**					
Code	PMID20	Year of study				
Course teacher	pred. Divna Krpan prof.dr. sc. Marko Rosić	Credits (ECTS)	6,0			
Associate teachers		Type of instruction (number of hours)	L	S	E	F
			30		30	
Status of the course		Percentage of application of e-learning				

COURSE DESCRIPTION

Course objectives	Use the knowledge acquired in this domain from previous education. Understand and learn procedures and activities for solving problems and developing computer software. Understand, acquire and learn the basic object oriented programming concepts. Understand, integrate and learn the basic concepts of storing and re-using dana
Course enrolment requirements and entry competences required for the course	Course enrolment requirements: none. Entry competences: basic knowledge of computer science.
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	1. Classify data types 2. Classify basic algorithm structures 3. Classify basic exceptions (errors, bugs) 4. Identify programming exceptions 5. Write code for handling exceptions 6. Write console and graphic user interface (GUI) applications in C# 7. Write user defined data types (structs) 8. Write class (properties, methods, constructors) 9. Identify data structures (simple and complex data types)
Course content broken down in detail by weekly class schedule (syllabus)	Week 1: Labs: Pre-test. Lectures: Course overview. Intorduction to C# programming language. Week 02: Labs: Basic input/output. Lectures: Software developing metodology, integrated developing environment (IDE), sample IDEs, IDE installation and setup. Week 03: Labs: Random number generation, minimum/maximum algorithms, prime number algorithm. Lectures: Data types in C# (simple: text and numeric), basic algorithm structures, decisions and loops. Week 04: Labs: Arrays (input, output), string arrays, two-dimensional arrays (matrix), methods Lectures: Complex data structures (one-dimensional and two-dimensional), structs Week 05: Labs: recursion Lectures: recursion, top-down method by example Week 06: Labs: mid-term exam preparation

<table>
<tr><td></td><td colspan="6">Lectures: software testing and debugging, exception types, preparation for mid-term exam
Week 07:
Labs: Mid-term exam 1
Lectures: introduction to graphical user interface (GUI), introduction to GUI .NET elements, controls
Week 08:
Labs: Developing simple GUI
Lectures: GUI developing environment
Week 09:
Labs: read/write data from different controls (for example: text controls into combo/list controls)
Lectures: Class and objects in C# (theory and syntax)
Week 10:
Labs: Applications with multiple forms, menu strip
Lectures: Advanced GUI interface elements.
Week 11:
Labs: Using and programming classes, instances (constructors, types, properties).
Lectures: Arrays of complex types, lists and collections.
Week 12:
Labs: Use of arrays and lists.
Lectures: Streams and files.
Week 13:
Labs: Assignments with streams and files.
Lectures: Advanced file storing systems (binary files).
Week 14:
Labs: preparation for second exam.
Lectures: Preparation for second exam
Week 15:
Labs: Final exam.
Lectures: Exam analysis</td></tr>
<tr><td>Format of instruction</td><td colspan="3">☒ lectures
☐ seminars and workshops
☒ exercises
☐ on line in entirety
☐ partial e-learning
☐ field work</td><td colspan="3">☒ independent assignments
☐ multimedia
☐ laboratory
☐ work with mentor
☐ homework assignments</td></tr>
<tr><td>Student responsibilities</td><td colspan="6">Attendance, active participation in the learning process, midterm exam, final exam, oral exam.</td></tr>
<tr><td rowspan="5">Screening student work <i>(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)</i></td><td>Name</td><td>Ects</td><td>Name</td><td>Ects</td><td>Name</td><td>Ects</td></tr>
<tr><td>Class attendance</td><td>1</td><td>Research</td><td></td><td>Experimental work</td><td></td></tr>
<tr><td>Oral exam</td><td>1</td><td>Report</td><td></td><td>Homework assignments</td><td></td></tr>
<tr><td>Seminar essay</td><td></td><td>Essay</td><td></td><td></td><td></td></tr>
<tr><td>Tests</td><td>1</td><td>Practical training</td><td>1</td><td></td><td></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>Written exam</td><td>2</td><td>Project</td><td></td><td></td><td></td></tr></table></td></tr>
<tr><td>Grading and evaluating student work in class and at the final exam</td><td>Practical exam consists of two parts, first (midterm exam) represents 40% of the final practical exam grade, and second part represents 60% of the practical exam grade. Students that fail at one or both parts of the practical exam during semester only write part which they did not pass. Everyone must also pass the oral exam which is 20% of the final grade.</td></tr>
<tr><td>Required literature (available in the library and via other media)</td><td><table>
<tr><th>Title</th><th>Number of copies in the library</th><th>Availability via other media</th></tr>
<tr><td>Griffiths, I., Adams, M., & Liberty, J. (2010). Programming C# 4.0: O'Reilly Media, Inc.</td><td>0</td><td></td></tr>
<tr><td>Online lecture notes and course materials</td><td>0</td><td></td></tr>
</table></td></tr>
<tr><td>Optional literature (at the time of submission of study programme proposal)</td><td></td></tr>
<tr><td>Quality assurance methods that ensure the acquisition of exit competences</td><td>Talk with students, student evaluation using the anonymous survey, the success of students in the exam, self-assessment.</td></tr>
<tr><td>Other (as the proposer wishes to add)</td><td></td></tr>
</table>