<table>
<tr><td>NAME OF THE COURSE</td><td colspan="5">Programming I</td></tr>
<tr><td>Code</td><td>PMID10</td><td>Year of study</td><td colspan="3">UGU-1</td></tr>
<tr><td>Course teacher</td><td>doc.dr. sc. Ani Grubišić</td><td>Credits (ECTS)</td><td colspan="3">6,0</td></tr>
<tr><td rowspan="2">Associate teachers</td><td rowspan="2">Ines Šarić
Ivan Bilobrk prof.
Mensur Duraković</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></td><td>30</td><td></td></tr>
<tr><td>Status of the course</td><td>mandatory</td><td>Percentage of application of e-learning</td><td colspan="3">10%</td></tr>
<tr><td colspan="6">COURSE DESCRIPTION</td></tr>
<tr><td>Course objectives</td><td colspan="5">Repeat the knowledge gained in this area in previous education. Understand, integrate and learn procedures and activities needed to solve problems and develop software on your computer. Understand, integrate and learn the basic programming concepts from the standpoint of program instructions for data acquisition, data processing, storage and distribution of the data processing results. Understand, integrate and learn the basic concepts of storing and re-using data.</td></tr>
<tr><td>Course enrolment requirements and entry competences required for the course</td><td colspan="5">Course enrolment requirements: none.
Entry competences: basic knowledge of 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 to:
1. classify the basic algorithmic structures
2. classify the main types of errors
3. identify bugs in software solution
4. make a flowchart and pseudocode of algorithm
5. compare basic sorting algorithms
6. write programs in Python
7. assess the validity of software solutions
8. establish the existence of an error in the software solution
9. evaluate software solutions</td></tr>
<tr><td>Course content broken down in detail by weekly class schedule (syllabus)</td><td colspan="5">Week 1:
Lectures: Introductory lecture: teachers, student obligations, elements of the student progress monitoring, examination, evaluation, Introduction to Programming: presentation of the course objectives, literature
Exercise: Log on Moodle, installation of Python development environment (Python Shell), Arithmetic Operators
Week 2:
Lectures: Development environment of programming language Python, simple data types: integers, real numbers, logic, strings, variables: naming variables, initialization of variables, constants, arithmetic operators, arithmetic expressions, comparison operators, logical operators, logical expressions, Assigning values, Changing the values of variables, variables and expressions, variables and strings, Python: PRINT, INPUT, formatted print,
Exercises: Types of variables, Print and Input commands
Week 3:
Lectures: The algorithms in general: history, characteristics, method of gradual refinement, flow chart and pseudocode, Algorithm structures, algorithms - linear structures, algorithms - branched structures: one selection, double-sided selection, multiple selection, Python: IF-THEN
Exercises: Logical and relational operators, IF command
Week 4:</td></tr>
</table>

	Lectures: Built-in (over) functions, Python functions for working with strings, operations for working with strings, conversion functions, functions with numbers, math functions Exercises: Built-in functions, complex numbers, strings - built-in functions Week 5: Lectures: Algorithms - cyclic structure: loop with a known number of repeats, the loop with condition at the beginning, the loop with condition at the end Exercises: For and while loop Week 6: Lectures: Python: for, while, a random number generator Exercise: Prime numbers, number digits, conversion of numerical system Week 7: Lectures: Procedures: subroutines (subroutine) and functions, local and global variables, Recursion, known recursive algorithms (factorial, Fibonacci numbers, 8 Queen, Hanoi Towers, Euclid's procedure), Python: def, return Exercises: Strings, Functions Week 8: Lectures: preparation for the colloquium Exercise: preparation for the colloquium Week 9: Lectures: Colloquium Exercise: Analysis of the colloquium or more excersises Week 10: Lectures: Sequences Exercises: Sequences Week 11: Lectures: Sorting: bubble sort, selection sort, insertion sort, quick sort Exercise: Sorting Week 12: Lectures: Data Files: definition, structure, physical and logical organization Typical processes for data processing, Python files: open, close, write, read, input, print Exercise: Complex problems with arrays Week 13: Lectures: syntax error, semantic and logical errors, errors in execution, the debugger (debugger), methods to detect errors: line by line, tracing from the point of interruption, observation, analysis of changes in the values of variables, steps in eliminating errors, categorizing problems , Python debugger Exercises: Files Week 14: Lectures: Programming, instructions, Programming, software (system and application), Programming languages: machine, assembler, high level programming languages, Compiler, interpreters, programming paradigms: procedural and non-procedural, structured and unstructured, functional, logical, object-oriented Exercises: Files Week 15: Lectures: The stages of software development, mathematical and physical model Exercises: Colloquium

<table>
<tr><td>Format of instruction</td><td>☒ lectures
☐ seminars and workshops
☐ exercises
☐ on line in entirety
☒ partial e-learning
☐ field work</td><td>☐ independent assignments
☐ multimedia
☐ laboratory
☐ work with mentor
☐ homework assignments</td></tr>
<tr><td>Student responsibilities</td><td colspan="2">Attendance, active participation in the learning process, homework, colloquium, exam</td></tr>
<tr><td>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)</td><td colspan="2">
<table>
<tr><th>Name</th><th>Ects</th><th>Name</th><th>Ects</th><th>Name</th><th>Ects</th></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>0,5</td><td>Report</td><td></td><td>Homework assignments</td><td>1</td></tr>
<tr><td>Seminar essay</td><td></td><td>Essay</td><td></td><td></td><td></td></tr>
<tr><td>Tests</td><td>2</td><td>Practical training</td><td>1</td><td></td><td></td></tr>
<tr><td>Written exam</td><td>0,5</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 colspan="2">Activity of students in lectures and exercises (attendance, problem solving, general activity in the classroom) (20%).
Written exam (50%): two colloquiums that are scored on a scale from 0 to 100 points. Students who achieve at least 100 points from both colloquiums are released from the written exam. Other students take written exam.
Oral exam (30%).
The final grade is derived on the basis of all the above ratings.</td></tr>
<tr><td>Required literature (available in the library and via other media)</td><td colspan="2">
<table>
<tr><th>Title</th><th>Number of copies in the library</th><th>Availability via other media</th></tr>
<tr><td>Budin, L., Brođanac, P., Markučić, Z., Perić, S. (2012) Rješavanje problema programiranjem u Pythonu, Element, Zagreb, ISBN: 978-953-197-395-3</td><td>15</td><td></td></tr>
</table>
</td></tr>
<tr><td>Optional literature (at the time of submission of study programme proposal)</td><td colspan="2">Griffiths, D., Barry, P. (2009) Head First Programming: A Learner's Guide to Programming Using the Python Language, ISBN: 978-0596802370
Teaching materials available on the Internet, including solution of selected tasks and additional scientific literature.</td></tr>
<tr><td>Quality assurance methods that ensure the acquisition of exit competences</td><td colspan="2">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 colspan="2"></td></tr>
</table>