

|      |                           |   |   |   |   |           |
|------|---------------------------|---|---|---|---|-----------|
| I056 | Obligatory,<br>I semester | <b>Introduction to Computer Science</b> | L | S | P | ECTS<br>6 |
|      |                           |   | 2 | 0 | 2 |           |

**Course objectives.** Course objectives are to familiarize students with basic ideas and methods of computer science and programming languages. Emphasis will be put on the basics of procedural programming in dynamic programming language (such as Python) and static programming language C/C++. Develop students' way of thinking that enables addressing more complex algorithmic and software solutions. During exercises, students will master the basic techniques of programming and learn how to deal with programming problems ranging from elementary to more complex.

**Course prerequisites.** Not required.

### Syllabus.

1. High-level, general-purpose, interpreted, dynamic programming language. Operators and variables. Lists. Tuples. Strings. Dictionaries. Branching. Loops. Functions.
2. Representation of numbers (integer and rational). Elementary arithmetic in different number systems. Error analysis. Significant digits. Floating-point arithmetic. Errors in calculating of the function values.
3. Source code. Compiler. Object code. Statements. Boolean expressions. Conditions.
4. Repetitions or loops. Functions and subroutines. Recursive functions. Local and global variables. Arrays. Pointers. Dynamic memory allocation.
5. Stack. Debugging. File I/O. Exceptions. Iterators.
6. Algorithms for some elementary problems: The problem of determining the maximum / minimum number. Search problem (binary search). Sorting problem. Recursive sorting. Matrix multiplication. Recursive matrix multiplication.
7. The basic idea of algorithm complexity.
8. Program testing (unittest)

### EXPECTED LEARNING OUTCOMES

| No. | LEARNING OUTCOMES  |
|-----|--|
| 1.  | To demonstrate knowledge and understanding which can serve as the foundation for developing and application of original ideas. |
| 2.  | To apply knowledge, understanding and skills in a broad variety of problems in the field of computer science.                  |
| 3.  | To integrate new knowledge to successfully solve programming problems in programming languages Python and C/C++.               |
| 4.  | To be able to present conclusions and findings to experts and laymen based on the knowledge and experience.                    |
| 5.  | To apply the acquired skills onto further education in this field.   |

**COUPLING OF THE EXPECTED LEARNING OUTCOMES, TEACHING PROCESS ORGANIZATION AND THE EVALUATION OF THE TEACHING OUTCOMES**

| TEACHING PROCESS ORGANIZATION | ECTS | EXPECTED LEARNING OUTCOMES ** | STUDENT ACTIVITY *  | EVALUATION METHOD   | SCORE |     |
|-------------------------------|------|-------------------------------|---|---|-------|-----|
|                               |      |                               |   |   | min   | max |
| Lecture attendance            | 1    | 1-5                           | Class attendance, discussion, solving the problems individually and in a team | Lists with signatures, observing the activity during the lectures | 0     | 4   |
| Homework                      | 1    | 1-4                           | Solving the problems individually   | Grading   | 12    | 20  |
| Repeated exams                | 2    | 1-4                           | Preparation for the written exam  | Grading   | 19    | 38  |
| Final exam                    | 2    | 1-4                           | Revising  | Oral exam   | 19    | 38  |
| TOTAL                         | 6    |                               |   |   | 50    | 100 |

**Teaching methods and student assessment.** Lectures will contain many examples with in-depth explanations. Exercises will be held in specialized computer-based laboratories where students will learn how to program in Python and C/C++. Final exam will be held after the completion of lectures and exercises and it will contain practical tasks each student will have to complete independently.

**Can the course be taught in English:** Yes

**Basic literature:**

1. D. Matijević, N. Truhar, Uvod u računarstvo, available on the official web site.
2. materials available at the web pages of Department of mathematics, University of Osijek

**Recommended literature:**

1. M. L. Hetland, Beginning Python: From Novice to Professional, Apress, 2008.
2. J. G. Brookshear, D. Brylow: Computer Science – An Overview (12 ed), Pearson Education, 2015.
3. Malan, D.J., Introduction to Computer Science, Harvard online course
4. J. Šribar, B. Motik: Demistificirani C++, Element, 2010. (3. prošireno izdanje).