COURSE DESCRIPTION

HIGHER VOCATIONAL STATE SCHOOL IN WLOCLAWEK

Course: ALGORITHMS AND COMPUTATIONAL COMPLEXITY

Field of study:	Computer Science	**Course code:**
Unit supervising the course:	Department of Computer Science	
Course orientation:	PRACTICAL	
Language of instruction:	English	
Course type:	General	
Course status:	Mandatory	
Level:	**Year:**	**Semester:**

The number of teaching hours on the full-time programme:

Total	**lecture**	**classes**	**laboratory**	**projects**	**tutorials**	**seminars**	**practicum**
30	**15**	-	**15**	-	-	-	-

The number of teaching hours on the part-time programme:

Total	**lecture**	**classes**	**laboratory**	**projects**	**tutorials**	**seminars**	**practicum**
-	-	-	-	-	-	-	-

Learning outcomes:	**Knowledge:** The student has a basic knowledge of the operation, description and analysis of the complexity of the algorithms required to self-solve algorithmic problems and implementation of their solutions using high-level languages. (K_W03)
	Skills: The student is able to plan and carry out experiments, including computer simulations, and is able to interpret the results and draw conclusions. (K_U02)
	Social competence: The student has a sense of responsibility for their own work and the willingness to submit to the work rules in a team and to take responsibility for collaborative tasks. (K_K04)

<table>
<tr><td>Full description of the course:</td><td colspan="5">Lecture program:

- definition of the algorithm, ways of describing algorithms, the concept of computational and memory complexity, average and worst-case analysis, asymptotic notation;

- simple data types, variables, basic programming language statements: conditional statements, loops, case and switch statements;

- complex data types: non-dynamic tables and records, examples of algorithms: linear and binary data search, simple sorting algorithms;

- functions, the concept of recursion, recursive algorithms: calculation of fractional and Fibonacci numbers, floodfill algorithm, recursive vs. iterative algorithms;

- pointers, dynamic data structures: stack, singly and doubly linked lists, cyclic linked lists, trees.

Laboratory exercises:

The implementation of laboratory exercises using Java and C programming languages coinciding with the topics of the lecture.</td></tr>
<tr><td>Methods:</td><td colspan="5">Lecture:
Lecture and multimedia presentation.

Laboratory exercises:
Implementation of algorithms in high-level programming languages.</td></tr>
<tr><td rowspan="4">The student’s workload/ ECTS credits:

1) included in the teaching load:
- lectures (15)
- laboratory exercises (15)

2) not included in the teaching load:
- personal consultation (6)
- e-mail consultation (6)
- course credit (6)</td><td rowspan="2">Forms of activities</td><td colspan="4">Average number of hours to complete activities</td></tr>
<tr><td colspan="2">Full-time</td><td colspan="2">Part-time</td></tr>
<tr><td>Lecture</td><td>Classes</td><td>Lecture</td><td>Classes</td></tr>
<tr><td>Contact hours with academic instructor</td><td>25</td><td>25</td><td>-</td><td>-</td></tr>
<tr><td>Hours without academic instructor</td><td>50</td><td>50</td><td>-</td><td>-</td></tr>
</table>

	1. Preparation for the classes, including reading assignments	25	15	-	-
	2. Processing the quantitative data /preparation for the exam, evaluation tests, etc.	25	20	-	-
	3. Preparation of a report, presentation, discussion	-	15	-	-
	Total	**75**	**75**	-	-
	Total number of ECTS for the conducted form of classes	**3**	**3**		
	Total number of ECTS points for the entire course	**6**			
The type and mode of obtaining the credit and marking criteria or requirements:	**The type:** Lecture – credit with a mark. Laboratory exercises – credit with a mark.				
	The mode: Lecture – one lecture test. Laboratory exercises – implementation of algorithms as separate programs. Presentation of programs.				
	Basic assessment criteria: Lecture – test of the material presented during lectures for positive mark. Laboratory exercises – realization of all appointed exercises each at least for positive mark.				
Literature:	**Prescribed reading:** [1] N. Wirth, "Algorithms and Data Structures", Prentice Hall, 1985. [2] A. V. Aho, J. D. Ullman, J. E. Hopcroft, "Data Structures and Algorithms", Addison Wesley Publishing Company, 1983. [3] G. Brassard, P. Bratley, "Algorithms. Theory & practice", Prentice Hall, Englewood Cliffs, New Jersey, 1988. **Recommended reading:** [1] B., R. Bravaco "Programming in Java. From the Ground				

	Up", McGraw-Hill Education – Europe, 2009. [2] E. Bruce, "Thinking in C++", vol. 1, 2, Prentice Hall, 2000.
Course instructor: **Dariusz Puchala, PhD**	