Course Syllabus

Course from study programme for the cycle: 2022/2023

I. General Information

Course name	Algorithms and computational complexity
Programme	Informatics
Level of studies (BA, BSc, MA, MSc, long-cycle MA)	BA
Form of studies (full-time, part-time)	full-time
Discipline	Informatics
Language of instruction	english

Course coordinator	Paweł Wójcik, PhD

Type of class *(use only the types mentioned below)*	Number of teaching hours	Semester	ECTS Points
lecture	15	VI	3
tutorial			
classes			
laboratory classes	15	VI	
workshops			
seminar			
introductory seminar			
foreign language classes			
practical placement			
field work			
diploma laboratory			
translation classes			
study visit			

Course pre-requisites	Basic programming knowledge (data types, conditional statements, loops, arrays)

II. Course Objectives

The aim of the course is to introduce students to issues related to design and analysis of algorithms and computational complexity.

III. Course learning outcomes with reference to programme learning outcomes

Symbol	Description of course learning outcome	Reference to programme learning outcome
KNOWLEDGE		
W_01	Students acquire knowledge of basic concepts of algorithms and computational complexity	K_W01 K_W03 K_W06
W_02	Students acquire knowledge of techniques of algorithms designing	K_W01 K_W03 K_W06
W_03	Students acquire knowledge of basic concepts of algorithm analysis	K_W01 K_W03 K_W06
SKILLS		
U_01	Students are able to implements presented algorithms in the programming language	K_U04 K_U08 K_U09 K_U17 K_U22
U_02	Students are able to analyse and evaluate presented algorithms	K_U07 K_U08 K_U09 K_U22
SOCIAL COMPETENCIES		
K_01	Students are able to formulate an opinion regarding algorithms and computational complexity	K_K01
K_02	Students can work on his/her own and in a team designing and developing algorithms	K_K02

IV. Course Content

1. Induction.
2. Computational complexity of algorithms. Asymptotic notations.
3. Recurrent schemes implicit and explicit representation.
4. Review of algorithmic techniques. Method of "divide and conquer".
5. Dynamic programming and greedy method.

V. Didactic methods used and forms of assessment of learning outcomes

Symbol	Didactic methods *(choose from the list)*	Forms of assessment *(choose from the list)*	Documentation type *(choose from the list)*
KNOWLEDGE			
W_01	Conventional Lecture / Problem lecture	Exam / Written test	Evaluated test / written test
W_02	Conventional Lecture / Problem lecture	Exam / Written test	Evaluated test / written test
W_03	Conventional Lecture /	Exam / Written test	Evaluated test / written

	Problem lecture		test
SKILLS			
U_01	Laboratory Classes / PBL (Problem-Based Learning) Design thinking	Exam / Written test	Evaluated test / written test
U_02	Laboratory Classes / PBL (Problem-Based Learning) Design thinking	Exam / Written test	Evaluated test / written test
SOCIAL COMPETENCIES			
K_01	Laboratory Classes / PBL (Problem-Based Learning) Design thinking	Exam / Written test	Evaluated test / written test
K_02	Laboratory Classes / PBL (Problem-Based Learning) Design thinking	Exam / Written test	Evaluated test / written test

VI. Grading criteria, weighting factors.....

Exam – 100%.

90 – 100% excellent

80 – 89% very good

70 – 79% good

60 – 69% satisfactory

50 – 59% sufficient

- 50% fail

VII. Student workload

Form of activity	Number of hours
Number of contact hours (with the teacher)	**60**
Number of hours of individual student work	**30**

VIII. Literature

Basic literature
K. A. Ross, Ch. Wright, Discrete Mathematic - V edition, Pearson 2002 N. P. Grimaldi, Discrete and Combinatorial Mathematics, Pearson 2004
Additional literature
T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, The MIT Press, 2009