

Bialystok University of Technology									
Field of study	Computer Science							Degree level and programme type	Engineer's degree full-time programme
Specialization/ diploma path	---							Study profile	academic
Course name	Algorithms and Data Structures							Course code	FCS-00020
								Course type	obligatory
Forms and number of hours of tuition	L	C	LC	P	SW	FW	S	Semester	3
	30	30			30			No. of ECTS credits	6
Entry requirements	Discrete Mathematics (FCS-00054), Object Oriented Programming (FCS-00012),								
Course objectives	<p>The aim of the course is to develop the ability to assess the effectiveness of the algorithm and the design of effective solutions before the implementation phase of the algorithm. Students will be equipped with knowledge on: methods of determining/estimating the cost of computational algorithmic methods efficient algorithmic design, design methods efficient data structures computationally difficult problems. Student evolve, skills: design computationally efficient algorithms and data structures, assessing the effectiveness of the solutions, identifying problems and applying computationally difficult approximate solutions for these problems. The aim of the course is also to develop skills to communicate effectively in the field of engineering and scientific problems with representatives of other disciplines.</p>								
Course content	<p>Lecture</p> <ol style="list-style-type: none"> <li>1) Basic concepts regarding algorithms (correctness, complexity)</li> <li>2) Recursion as a technique of programming</li> <li>3) Techniques of developing algorithms: "divide and conquer", greedy and dynamic programming</li> <li>4) Implementations of dictionary structure (trees and hashtables)</li> <li>6) Graph data structures and graph algorithms</li> <li>7) Complexity classes (P, NP, NPC, NP-hard), examples of computationally hard problems</li> <li>8) Approximate algorithms and heuristics</li> </ol> <p>Excercises:</p> <ol style="list-style-type: none"> <li>1) Determining time complexity of computer programs, developing effective solutions for basic computational problems</li> <li>2) Practice regarding recursion and "divide and conquer (desing of algorithms, computing time complexity)</li> <li>3) Solving computational problems with greedy and dynamic programming techniques.</li> <li>4) Effective data structures (heaps, trees, hashtables)</li> <li>5) Solving graph problems</li> <li>6) Recognition of computationally hard problems and applying approximate/heuristic solutions.</li> </ol> <p>Computer labs:</p> <ol style="list-style-type: none"> <li>1) Basic computational problems, optimization of algorithms in terms of time complexity (comparison, design and implemantation)</li> <li>2) Solving computational problems with recursion/"divide and conquer" techniques.</li> <li>3) Design and implementation of efficient algorithms (greedy and dynamic programming techniques) solving combinatorial optimization problems.</li> <li>4) Implementation of tree data structures</li> <li>5) Solving graph problems.</li> </ol>								
Teaching methods	informative lecture, lecture problem, case method, programming, subject exercises,								
Assessment method	Lecture - written exam; exercise - test; Laboratory - assessment of project								
Symbol of learning outcome	Learning outcomes							Reference to the learning outcomes for the field of study	
LO1	knows the fundamental concepts and notations used in the analysis of algorithms and data structures and describing computing problems decision-making, etc., which can be solved by a computer.							K_W05 K_U01	
LO2	is able to assess the time and memory complexity of algorithms and data structures. He can compare different solustions to the same problem.							K_W05 K_U04	
LO3	knows the standard solutions of designing and implementation of algorithms and data structures, their properties and application areas.							K_W05	
LO4	can propose and design or choose algorithms and structures data to effectively solve a given engineering or scientific. Able to estimate the complexity of the problem and identify.							K_W05 K_U12	
LO5	is able to verify the correctness of algorithms using basic methods formal and simulation.							K_W05 K_U01 K_K03	
LO6	knows how to implement algorithms and data structures using high level programming languages							K_U05	
Symbol of learning outcome	Methods of assessing the learning outcomes							Type of tuition during which the outcome is assessed	
LO1	test in the context of tutorials, written exam							L, C	
LO2	test in the context of tutorials, written exam							L, C	
LO3	test in the context of tutorials, written exam							L	
LO4	credit specialist laboratory tasks							Sw	
LO5	test, written exam, credit specialist laboratory tasks							L,C,Sw	
LO6	credit specialist laboratory tasks, observation of work during the workshop							Sw	
Student workload (in hours)							No. of hours		
Calculation	1 - Participation in lectures -							30	
	2 - Participation in the workshop an specialist auditorium -							60	
	4 - Participation in the consultations related to laboratory exercises and spec. -							5	
	5 - Execution of the tasks of problem specialist workshop. -							35	
	6 - Exam preparation -							20	
	<b>TOTAL:</b>							<b>150</b>	
Quantitative indicators							HOURS	No. of ECTS credits	
Student workload - activities that require direct teacher participation							95 (4)+(2)+(1)	3.8	
Student workload - practical activities							95 (2)+(5)	3.8	

<b>Basic references</b>	1. Adam Drozdek, Data structures and algorithms in C++, 2013. 2. T.H. Cormen, Introduction to algorithms, 2001.	
<b>Supplementary references</b>	1. M.T. Goodrich, Roberto Tamassia, Data structures and algorithms in Java, 2001.	
<b>Organisational unit conducting the course</b>	Department of Theoretical Computer Science	<b>Date of issuing the programme</b>
<b>Author of the programme</b>	dr Joanna Karbowska-Chilińska, dr inż. Krzysztof Ostrowski	Feb. 17, 2022

L – lecture, C – classes, LC – laboratory classes, P – project, SW – specialization workshop, FW – field work, S – seminar