University of Gdańsk

Subject card

Subject name and code	Theoretical basis of computer science, PG_00143809		
Field of study	Informatics		
Date of commencement of studies	October 2025	Academic year of realisation of subject	2026/2027
Education level	Bachelor's studies	Subject group	Obligatory subject group in the field of study Subject group related to scientific research in the field of study
Mode of study	part-time studies	Mode of delivery	at the university
Year of study	2	Language of instruction	Polish
Semester of study	4	ECTS credits	3.0
Learning profile	academic	Assessment form	
Conducting unit			
Name and surname of lecturer (lecturers)	Subject supervisor	dr Hanna Furmańczyk	
	Teachers		

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	20.0	0.0	0.0	0.0	0.0	20

E-learning hours included: 0.0

Learning activity and number of study hours

Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
Number of study hours	20	0.0	55.0	75

Subject objectives

The aim of the course is to familiarize students with the theoretical foundations of computer science - automata, languages and the basics of computational complexity.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFL3_U01] can apply mathematical knowledge to formulate, analyze and solve problems related to computer science	can determine the class of formal languages can prove that a language belongs to a selected class can apply acquired knowledge to determine the decidability of simple problems	[SU3] text preparation/written work [SU8] observation of student's independent or team work
[INFL3_W03] has structured, theoretically grounded general knowledge in the field of algorithms and data structures, formal languages, automata theory and computational complexity, and artificial intelligence	knows the basics of formal languages knows definitions and examples of regular expressions, finite automata, context-free grammars, stacked automata, Turing machines knows Chomsky's hierarchy knows the definition and examples of decidable and undecidable problems	[SW4] test/exam - oral or written
[INFL3_U03] can design and analyze algorithms for their correctness and computational complexity using appropriate algorithmic techniques and data structures	is able to analyze known algorithms in terms of their computational complexity	[SU4] test/exam - oral or written [SU8] observation of student's independent or team work

<table>
<tr><td>Subject contents</td><td>1. Finite automata, regular expressions, non-deterministic automata, equivalence of regular expressions and finite automata.
2. Regular expressions in scripting languages.
3. Chomsky's hierarchy. Context-free grammars, automata with stack, derivation trees, parsers.
4. Turing machine as a model of computation. The concept of computability.
5. Computational complexity.
6. Class P and NP. NP-complete problems.</td></tr>
<tr><td>Prerequisites and co-requisites</td><td>no requirements</td></tr>
<tr><td>Assessment methods and criteria</td><td>
<table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>test</td><td>51.0%</td><td>60.0%</td></tr>
<tr><td>homework</td><td>0.0%</td><td>20.0%</td></tr>
<tr><td>activity on lectures</td><td>0.0%</td><td>20.0%</td></tr>
</table>
</td></tr>
<tr><td>Recommended reading</td><td>
<table>
<tr><td>Basic literature</td><td>Hopcroft, Ullman, Wprowadzenie do teorii automatów, języków i obliczeń
J. Jędrzejowicz, A. Szepietowski, Języki, automaty, złożoność obliczenio
Wyd. UG 2008</td></tr>
<tr><td>Supplementary literature</td><td>no applicable</td></tr>
<tr><td>eResources addresses</td><td>Adresy na platformie eNauczanie:</td></tr>
</table>
</td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>no applicable</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

Document generated electronically. Does not require a seal or signature.