GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Basics of programming - algorithms and data structures, PG_00062716		
Field of study	Technologies for Industry 5.0		
Date of commencement of studies	October 2025	Academic year of realisation of subject	2025/2026
Education level	first-cycle studies	Subject group	Obligatory subject group in the field of study
Mode of study	Full-time studies	Mode of delivery	at the university
Year of study	1	Language of instruction	Polish
Semester of study	1	ECTS credits	6.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Department Of Biomedical Engineering -> Faculty Of Electronics Telecommunications And Informatics -> Wydziały Politechniki Gdańskiej		
Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Grzegorz Jasiński	
	Teachers	dr inż. Grzegorz Jasiński	

Lesson types and methods of instruction

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	30.0	0.0	30.0	0.0	0.0	60

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	60	5.0	85.0	150

Subject objectives

The aim of the course is for students to acquire knowledge and skills in programming. The student should master the ability to create and analyse algorithms and the principles of programming in the C/C++ language: instructions, data types, operators and functions. Students should acquire knowledge of structures, pointers and other basic concepts related to programming in C/C++.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_K01] is aware of the need to constantly update and enrich knowledge and practical skills, and improve professional, personal and social competences	The student understands the need to explore and learn well-known technologies. The student is able to analyze problems and select appropriate data models and data structures. The student understands the life cycle and development of code.	[SK2] Assessment of progress of work [SK5] Assessment of ability to solve problems that arise in practice [SK3] Assessment of ability to organize work
[K6_W01] demonstrates knowledge and understanding of mathematics, physics, chemistry and IT tools at the level necessary to formulate and solve typical engineering and technological problems	The student knows basic data structures and corresponding algorithms. The student is able to write programs in C/C++ using appropriate libraries.	[SW1] Assessment of factual knowledge
[K6_U01] applies knowledge of mathematics, physics, chemistry, IT tools and other engineering disciplines to solve theoretical, engineering and technological problems	The student programs in a procedural programming language, runs and tests programs. The student is able to analyze problems and select appropriate data models and structures data.	[SU2] Assessment of ability to analyse information [SU4] Assessment of ability to use methods and tools

<table>
<tr><td>Subject contents</td><td>1. Programming languages, alphabet, syntax and semantics. Translation.
2. Classification of types. Integer and floating point types.
3. Arithmetic operators and expressions.
4. Selected standard functions.
5. Character type. Type casting.
6. Logical type. Logical operators and expressions.
7. Basics of input/output.
8. Conditional statements (if, switch) and conditional expression.
9. Iterative instructions (for, while, do-while). Nested iterations.
10. Defining types. Constant. Enum type.
11. One-dimensional and multidimensional arrays. Strings.
12. Validity scope and lifetime of variables
13. Functions.
14. Function parameters.
15. Pointer type. Pointer arithmetic.
16. Pointers in communication between functions.
17. Dynamic memory allocation.
18. Structures
19. The concept of an algorithm, differences between algorithms, the influence of the model on the solution, representation of algorithms, correctness analysis.
20. Examples of recursive and iterative solutions, exhaustive and heuristic methods, dynamic programming.
21. Examples of recursive divide and conquer solutions.
22. Basic data structures: list, queue, stack and methods of their implementation.
23. Basic sorting algorithms, binary search, quick sort, heap sort.</td></tr>
<tr><td>Prerequisites and co-requisites</td><td></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>Lecture</td><td>50.0%</td><td>50.0%</td></tr>
<tr><td>Lab</td><td>50.0%</td><td>50.0%</td></tr>
</table></td></tr>
<tr><td>Recommended reading</td><td><table>
<tr><td>Basic literature</td><td>KERNIGHAN, Brian W.; RITCHIE, Dennis M. The C programming language, Prentice Hall, 2006

Gręboszc Jerzy, Symfonia C++ Standard (tom 1 i 2), Wydanie 2000, Krakow 2008</td></tr>
<tr><td>Supplementary literature</td><td>Stephen Prata, "Jezyk C++. Szkoła programowania". Wydanie VI. Helion 2012

Mirosław J. Kubiak, "C++. Zadania z programowania z przykładowymi rozwiazaniami", Helion 2011</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>Writing a programme that implements the given functionality.
Analyse how the given programme works.</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

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