GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Basics of Programming, PG_00047377
Field of study	Electronics and Telecommunications

Date of commencement of studies	October 2021	Academic year of realisation of subject	2021/2022
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	5.0
Learning profile	general academic profile	Assessment form	assessment

Conducting unit	Department of Algorithms and Systems Modelling -> Faculty of Electronics, Telecommunications and Informatics

Name and surname of lecturer (lecturers)	Subject supervisor	dr Paweł Obszarski
	Teachers	dr Paweł Obszarski
		dr inż. Joanna Raczek
		dr inż. Marcin Strąkowski
		dr inż. Andrzej Marczak
		dr hab. inż. Robert Janczewski
		mgr inż. Kacper Wereszko
		dr inż. Marek Tatara
		dr hab. inż. Iwona Kochańska
		dr inż. Maciej Wróbel
		mgr inż. Andrzej Jastrzębski
		mgr inż. Jan Glinko

Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	15.0	20.0	0.0	65
	E-learning hours included: 0.0						
	Adresy na platformie eNauczanie:						

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	65	10.0	50.0	125

Subject objectives	This course is an introduction to computer programming. Its primary objective is to teach solving of programming problems and writing programs using the C++ programming language.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study	The student analyzes and solves basic programming problems. The student uses high-level programming language constructs that are syntactically and semantically correct. He writes and modifies programs in a high level language. Identifies and removes the causes of program malfunctions. He explains and can apply basic programming techniques.	[SU1] Assessment of task fulfilment [SU4] Assessment of ability to use methods and tools
	[K6_U05] can plan and conduct experiments related to the field of study, including computer simulations and measurements; interpret obtained results and draw conclusions	Student writes simple programs to support the process of solving problems related to the field of his studies.	[SU1] Assessment of task fulfilment
	[K6_W04] Knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	The student knows the constructs of a high-level programming language. He is able to interpret and analyze programs written in a high-level programming language. The student knows the rules for selection of programming methods and language constructs.	[SW1] Assessment of factual knowledge

Subject contents	
	1. Introduction. 2. Programming languages, alphabet, syntax and semantics. Translation. 3. Classification of data types. Integer and floating point types. Arithmetic expressions and operators. Selected standard mathematical functions. 4. Character type. Casting of types. 5. Logical type. Logical operators and expressions. 6. Input/output basiscs. 7. Conditional statements (if, switch) and conditional expression. 8. Iteration statements (for, while, do-while), nested iterations. 9. Defining types. Constants. Enumerated type. 10. One- and multi-dimensional arrays. Null-terminated strings. 11. Basic rules for scope and lifetime of variables. 12. Functions. Scope and lifetime of variables. Side effect. 13. Passing parameters of a function. 14. Pointer type and pointer arithmetics. 15. Pointers for inter-function communication. 16. Dynamic memory allocation. 17. Basic dynamic data structures. 18. Structures (records). 19. Data structures using records and their applications. 20. Applications of dynamic data structures (stacks, queues, graph structures) 21. Introduction to object oriented programming. 22. Defining classes (constructor, destructor, data and function members). 23. Applications object-oriented programming paradigm. 24. Input/output streams classes. Input/output formatting. 25. Processing files. 26. Applications of recurrence (e.g., divide and conquer, greediness, dynamic programming).

Prerequisites and co-requisites	
	No requirements

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Programming lab assignments (1-3)	50.0%	30.0%
	Project assignments (1-2)	50.0%	30.0%
	Final test	50.0%	40.0%

Recommended reading	Basic literature	[1] Grębosz J., Symfonia C++ Standard (tom 1 i 2), Editions 2000, Krakow 2008.
		[2] Borowiecki Piotr, Podstawy programowania - materiały do wykładu.
		[3] Materiały przygotowujące do laboratorium z Podstaw programowania (opracowanie zespołowe).
		For participants of the course, [2] and [3] are available at eNauczanie PG platform.
	Supplementary literature	No requirements
	eResources addresses	
Example issues/ example questions/ tasks being completed		
Work placement	Not applicable	