University of Gdańsk

Subject card

Subject name and code	Programming, PG_00193434		
Field of study	Quantum Information Technology		
Date of commencement of studies	October 2026	Academic year of realisation of subject	2026/2027
Education level	Master's 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	English
Semester of study	1	ECTS credits	3.0
Learning profile	academic	Assessment form	credit
Conducting unit			
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. inż. Piotr Mironowicz	
	Teachers		

Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	15.0	0.0	0.0	30
	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	30	0.0	45.0	75

Subject objectives	The aim of this course is to provide a student a comprehensive overview of programming methodology that can be useful in further independent research in quantum information

Learning outcomes

Course outcome	Subject outcome	Method of verification
[QITL3_W02] knows and understands key topics and selected topics within the scope of advanced, detailed knowledge in the field of quantum information technologies.		
[QITL3_W01] knows and understands in depth selected facts, objects, and phenomena, as well as the methods and theories explaining the complex relationships between them, constituting advanced general knowledge in the field of quantum information technologies.		

Subject contents	Review and systematics of programming languages. Imperative and declarative programming. History and labor market. Programming environments.Program structure in C ++, Python, Matlab.Basic constructions. Variables, loops, conditional statements, functions, I / O operations, operators.Object-oriented programming. Classes. Basic data structures. Array, list, heap, map, graph.Code organization. Comments, headers, libraries, naming conventions. Programming Pragmatics. Programming styles. Version control systems.Doxygen.Recursion. Dynamic programming. Basic algorithms. Searching, sorting, graph searching.STL library in C ++. Design patterns. Processes and threads. Multi-threaded programming. Data Representations. XML. Sparse matrices. COO andCRS formats.Functional programming.Numerical Methods. Newton-Raphson method, Simpson method, Runge-Kutta method, matrix decompositions.Numpy and scipy packages in Python. Matlab QETLAB package.Linear and semi-definite programming. Solvers.Computational models. Turing machine. Church's thesis. Computational and memory complexity of algorithms. Complexity classes P, NP, NPC,PSPACE. Compilation process and parameters. Debugging and profiling. Unit tests.Code optimization techniques. Language interoperability. MEX files in Matlab. Extension modules in Python.CISC and RISC architectures. Flynn taxonomy. MMX, SSE, AVX instruction sets. Programming on graphic cards. CUDA, PyTorch.Virtual machines and emulators. Bytecode in Python. Assembler and low-level code optimization.BPP, BQP, QMA complexity classes. Quantum programming languages
Prerequisites and co-requisites	None.

Assessment methods and criteria

Subject passing criteria	Passing threshold	Percentage of the final grade
laboratory part: test	51.0%	50.0%
lecture part: test	51.0%	50.0%

Recommended reading

Basic literature	None.
Supplementary literature	None.
eResources addresses	

Example issues/ example questions/ tasks being completed	
Work placement	Not applicable

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