

Karta przedmiotu

Nazwa i kod przedmiotu	Programming, PG_00193434						
Kierunek studiów	Quantum Information Technology (O)						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2026/2027		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			angielski		
Semestr studiów	1	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca							
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. Piotr Mironowicz				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach		Praca własna studenta		RAZEM
	Liczba godzin pracy studenta	30	0.0		45.0		75
Cel przedmiotu	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						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu			Sposób weryfikacji i oceny efektu	
	[QITL3_W02] zna i rozumie kluczowe zagadnienia oraz wybrane zagadnienia z zakresu zaawansowanej wiedzy szczegółowej z zakresu technologii informacji kwantowej.						
	[QITL3_W01] zna i rozumie w pogłębionym stopniu wybrane fakty, obiekty i zjawiska oraz dotyczące ich metody i teorie wyjaśniające złożone zależności między nimi, stanowiące zaawansowaną wiedzę ogólną z zakresu technologii informacji kwantowej.						

Treści przedmiotu	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 and CRS 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		
Wymagania wstępne i dodatkowe	None.		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	laboratory part: test	51.0%	50.0%
	lecture part: test	51.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	None.	
	Uzupełniająca lista lektur	None.	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.