



## Subject card

Subject name and code	Parallel Programming for Multi-Core Architecture, PG_00064482						
Field of study	Informatics						
Date of commencement of studies	February 2026		Academic year of realisation of subject		2026/2027		
Education level	second-cycle studies		Subject group		Optional subject group Specialty subject group Subject group related to scientific research 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	2		ECTS credits		4.0		
Learning profile	general academic profile		Assessment form		exam		
Conducting unit	Department Of Computer Architecture -> Faculty Of Electronics Telecommunications And Informatics -> Wydziały Politechniki Gdańskiej						
Name and surname of lecturer (lecturers)	Subject supervisor		dr hab. inż. Paweł Czarnul				
	Teachers		mgr inż. Robert Kałaska  dr hab. inż. Paweł Czarnul				
Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	15.0	15.0	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		8.0		32.0	100
Subject objectives	learning techniques of parallel programming and APIs allowing use of modern manycore platforms						

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K7_W04] knows and understands, to an increased extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or other elements or programmable devices specific to the field of study, and organization of work of systems using computers or such devices	student knows basic rules and techniques of multithreaded programming for multi-core architectures	[SW1] Assessment of factual knowledge
	[K7_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, making assessment and critical analysis of the prepared software as well as a synthesis and creative interpretation of information presented with it	student knows how to select appropriate APIs and optimization methods for applications running on multicore systems	[SU1] Assessment of task fulfilment
	[K7_W10] knows and understands, to an increased extent, the basic processes occurring in the life cycle of equipment, objects and technical systems, as well as methods of supporting processes and functions, specific to the field of study	knows processes and dependencies concerning execution of parallel applications in multi-core systems	[SW1] Assessment of factual knowledge
	[K7_U12] is able, to an increased extent, to analyze the operation of components and systems related to the field of study, as well as to measure their parameters and study their technical characteristics, and to plan and carry out experiments related to the field of study, including computer simulations, interpret the obtained results and draw conclusions	knows how to analyze and profile runs of parallel applications	[SU1] Assessment of task fulfilment
Subject contents	<ol style="list-style-type: none"> <li>1. Passing criteria</li> <li>2. Current HPC systems</li> <li>3. Goals of parallel programming</li> <li>4. GPU as a parallel compute device</li> <li>5. Data decomposition</li> <li>6. Data parallel algorithms</li> <li>7. CUDA programming model</li> <li>8. GPU architecture</li> <li>9. Threads in CUDA</li> <li>10. Memory access in CUDA</li> <li>11. Optimizations using CUDA</li> <li>12. Using many GPUs</li> <li>13. Application debugging</li> <li>14. Unified Memory</li> <li>15. OpenCL for GPUs/CPUs</li> <li>16. Multicore CPUs</li> <li>17. Many/multicore architectures</li> <li>18. OpenMP</li> <li>19. Offload, native, symmetric modes</li> <li>20. Optimization (load balancing, synchronization)</li> <li>21. Parallelization models for various paradigms in OpenMP.</li> <li>22. Vectorization</li> <li>23. False sharing</li> <li>24. Thread affinity</li> <li>25. Synchronization</li> <li>26. Optimization divide-and-conquer</li> <li>27. Optimization computing similarity of vectors</li> <li>28. CPU+GPU programming</li> </ol>		

Prerequisites and co-requisites	basic knowledge of parallel programming		
	C programming knowledge		
Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	colloquium 1 + 2	50.0%	20.0%
	exam	50.0%	30.0%
	project	50.0%	25.0%
	laboratories	50.0%	25.0%
Recommended reading	Basic literature	[1] Pawel Czarul. Parallel Programming for Modern High Performance Computing Systems. Taylor & Francis. 2018 ISBN 9781138305953	
		[2] CUDA C programming guide. NVIDIA	
		[3] OpenMP specification	
		[4] OpenCL specification	
	Supplementary literature	CUDA documentation - NVIDIA presentations	
	eResources addresses	Adresy na platformie eNauczanie:	
Example issues/ example questions/ tasks being completed			
Work placement	Not applicable		

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