



## Subject card

Subject name and code	Concurrent and parallel programming, PG_00060227						
Field of study	Technical Physics						
Date of commencement of studies	October 2026	Academic year of realisation of subject				2028/2029	
Education level	first-cycle studies	Subject group				Optional 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	3	Language of instruction				Polish	
Semester of study	5	ECTS credits				3.0	
Learning profile	general academic profile	Assessment form				assessment	
Conducting unit	Institute of Physics and Applied Computer Science -> Faculty of Applied Physics and Mathematics -> Faculties of Gdańsk University of Technology						
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. Jan Franz					
	Teachers						
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	30.0	0.0	0.0	45
	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	45		5.0		25.0	75
Subject objectives	The aim of the course is to introduce students to the principles and techniques of concurrent and parallel programming in Java, with particular emphasis on their applications to simple physical problems. Students learn to design, implement, and analyze multithreaded programs, understand classical synchronization problems and design patterns, and evaluate the performance and scalability of parallel solutions.						
Learning outcomes	Course outcome	Subject outcome			Method of verification		
	[K6_W05] has knowledge of programming methodologies and techniques, as well as the use of selected IT tools in physics and engineering.	knows the basic methods and techniques of concurrent and parallel programming in Java and understands ways of solving classical synchronization problems in the context of simple physical and technical applications.			[SW1] Assessment of factual knowledge		
	[K6_U03] possesses programming skills in a selected language and the ability to use selected software packages.	is able to create concurrent and parallel programs in Java and use basic tools and libraries supporting the implementation, testing, and analysis of multithreaded programs.			[SU1] Assessment of task fulfilment		
	[K6_K01] demonstrates readiness for continuous learning and updating knowledge in physics and related fields, critically evaluating it and recognising its importance in solving practical and theoretical problems.	is prepared to expand their knowledge of concurrent and parallel programming and to critically assess the appropriateness of applying these techniques to solve practical and cognitive problems.			[SK5] Assessment of ability to solve problems that arise in practice		

Subject contents	<p>Course content – lecture</p> <p><b>Lectures</b></p> <ol style="list-style-type: none"> <li>1. <b>Introduction and Motivation</b> Concurrency concepts motivated by physics simulations with many particles and events.</li> <li>2. <b>Thread Life Cycle</b> Thread creation, execution, and termination; control methods and runtime costs.</li> <li>3. <b>Synchronization Fundamentals</b> Race conditions, critical sections, and basic synchronization mechanisms.</li> <li>4. <b>Performance Scaling Laws</b> Parallel speedup, efficiency, and scaling laws in computational physics.</li> <li>5. <b>Locks and Condition Variables</b> Mutual exclusion and coordination for safe access to shared simulation data.</li> <li>6. <b>ProducerConsumer Pattern</b> Event generation and processing pipelines inspired by detector systems.</li> <li>7. <b>Case Study I: Parallel Data Analysis</b> Concurrent data collection and processing in physics experiments.</li> <li>8. <b>ReaderWriter Pattern</b> Managing concurrent access to shared measurement datasets.</li> <li>9. <b>Deadlock and Resource Management</b> Deadlock, livelock, and starvation in systems with shared resources.</li> <li>10. <b>Thread Pools and Executors</b> Task-based parallelism using executors, futures, and thread pools.</li> <li>11. <b>Advanced Synchronization Tools</b> Semaphores and barriers for coordinated parallel computations.</li> <li>12. <b>ForkJoin Framework</b> Divide-and-conquer algorithms for large-scale numerical problems.</li> <li>13. <b>Parallel Streams</b> Data-parallel operations and reductions for scientific computations.</li> <li>14. <b>Case Study II: Parallel Physics Simulations</b> Examples from diffusion models, lattice systems, and scattering problems.</li> <li>15. <b>Summary and Outlook</b> Overview of key concepts and outlook toward high-performance computing.</li> </ol>
	<p>Course content – laboratory</p> <p><b>Laboratory classes</b></p> <ol style="list-style-type: none"> <li>1. <b>Introduction to Threads</b> Create basic threads; simulate parallel motion of particles in free fall.</li> </ol>

	<p>2. <b>Parallel Harmonic Oscillators</b> Run multiple oscillators in parallel and compare execution times.</p> <p>3. <b>Shared Data Synchronization</b> Simulate radioactive decay with safe updates of a shared counter.</p> <p>4. <b>Parallel Speedup Tests</b> Monte Carlo integration with many threads; analyze speedup.</p> <p>5. <b>Locks and Conditions</b> Model random walkers with coordinated access to shared data.</p> <p>6. <b>Producer Consumer Pattern</b> Simulate a detector where events are generated and analyzed concurrently.</p> <p>7. <b>Parallel Histogramming</b> Construct particle energy histograms using thread safe structures.</p> <p>8. <b>Readers and Writers</b> Concurrent access to a shared set of measurement results.</p> <p>9. <b>Deadlock Scenarios</b> Demonstrate deadlock in competing processes and apply avoidance methods.</p> <p>10. <b>Executors and Futures</b> Parallel random walk simulations with collection of statistics.</p> <p>11. <b>Barrier Synchronization</b> Synchronize oscillators at fixed time steps to model collective motion.</p> <p>12. <b>Fork Join Algorithms</b> Parallel summation of large datasets from physical simulations.</p> <p>13. <b>Parallel Stream Processing</b> Compute and reduce particle energies in parallel.</p> <p>14. <b>Parallel Diffusion Model</b> Simulate two dimensional diffusion and study performance scaling.</p> <p>15. <b>Integrated Simulation Project</b> Combine decay simulation and histogramming; compare performance.</p>									
Prerequisites and co-requisites										
Assessment methods and criteria	<table border="1"> <thead> <tr> <th>Subject passing criteria</th> <th>Passing threshold</th> <th>Percentage of the final grade</th> </tr> </thead> <tbody> <tr> <td>Lecture: exam</td> <td>50.0%</td> <td>25.0%</td> </tr> <tr> <td>Laboratory: programming exercises</td> <td>50.0%</td> <td>75.0%</td> </tr> </tbody> </table>	Subject passing criteria	Passing threshold	Percentage of the final grade	Lecture: exam	50.0%	25.0%	Laboratory: programming exercises	50.0%	75.0%
Subject passing criteria	Passing threshold	Percentage of the final grade								
Lecture: exam	50.0%	25.0%								
Laboratory: programming exercises	50.0%	75.0%								

Recommended reading	Basic literature	<p>1. B. Wittman, T. Korb, A. Mathur, Start Concurrent: An Introduction to Problem Solving in Java with a Focus on Concurrency, Purdue University Press, Ashland, Oregon, 2014.</p> <p>2. T. Rauber, G. Rünger, Parallel Programming: for Multicore and Cluster Systems, Springer Nature, Berlin, 2010.</p> <p>3. S. Selikoff, J. Boyarsky, OCA/OCP Java SE 8 programmer: practice tests, Sybex, Indianapolis, Indiana, 2017.</p>
	Supplementary literature	<p>1. M. Ben-Ari, "Principles of Concurrent and Distributed Programming", 2nd edition, Addison Wesley, Upper Saddle River, NJ, 2006.</p> <p>2. R.-G. Urma, M. Fusco, A. Mycroft, Modern Java in Action, Manning Publications, Shelter Island, 2018.</p> <p>3. B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, D. Lea, Java Concurrency in Practice. Addison-Wesley, Upper Saddle River, NJ, 2006.</p> <p>4. B. J. Evans, J. Clark, M. Verburg, The Well-Grounded Java Developer, Second Edition, Manning Publications, Shelter Island, 2023</p>
	eResources addresses	
Example issues/ example questions/ tasks being completed	<p>1. Calculate the Speedup of a concurrent program.</p> <p>2. Convert a sequential program in a concurrent program.</p>	
Practical activities within the subject	Not applicable	

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