GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Concurrent and parallel programming, PG_00060227		
Field of study	Technical Physics		
Date of commencement of studies	October 2024	Academic year of realisation of subject	2026/2027
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

eNauczanie source addresses:

Moodle ID: 1308 Programowanie współbieżne i równoległe
https://enauczanie.pg.edu.pl/2025/course/view.php?id=1308

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 course introduces the principles and techniques of concurrent and parallel programming in Java, with special emphasis on their application in simple physics-related problems. Students learn to design, implement, and analyze multi-threaded programs, understand classical synchronization issues and design patterns, and evaluate performance and scalability of parallel solutions.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_K01] understands the need to learn and improve professional and personal competencies, inspires and organizes other people's learning process	The student knows the scenarios where the use of concurrent or parallel programming is necessary to solve the problem.	[SK5] Assessment of ability to solve problems that arise in practice
[K6_W05] has knowledge of programming methodology and techniques, and the use of selected IT tools in physics and technology	The student learns how to solve classical concurrency problems.	[SW1] Assessment of factual knowledge
[K6_U03] knows programming languages and can use basic software packages	The student practices concurrent programming using the Java programming language.	[SU1] Assessment of task fulfilment

Subject contents	Course content – lecture **Lectures** 1. **Introduction and Motivation** Concurrency concepts motivated by physics simulations with many particles and events. 2. **Thread Life Cycle** Thread creation, execution, and termination; control methods and runtime costs. 3. **Synchronization Fundamentals** Race conditions, critical sections, and basic synchronization mechanisms. 4. **Performance Scaling Laws** Parallel speedup, efficiency, and scaling laws in computational physics. 5. **Locks and Condition Variables** Mutual exclusion and coordination for safe access to shared simulation data. 6. **ProducerConsumer Pattern** Event generation and processing pipelines inspired by detector systems. 7. **Case Study I: Parallel Data Analysis** Concurrent data collection and processing in physics experiments. 8. **ReaderWriter Pattern** Managing concurrent access to shared measurement datasets. 9. **Deadlock and Resource Management** Deadlock, livelock, and starvation in systems with shared resources. 10. **Thread Pools and Executors** Task-based parallelism using executors, futures, and thread pools. 11. **Advanced Synchronization Tools** Semaphores and barriers for coordinated parallel computations. 12. **ForkJoin Framework** Divide-and-conquer algorithms for large-scale numerical problems. 13. **Parallel Streams** Data-parallel operations and reductions for scientific computations. 14. **Case Study II: Parallel Physics Simulations** Examples from diffusion models, lattice systems, and scattering problems. 15. **Summary and Outlook** Overview of key concepts and outlook toward high-performance computing.
	Course content – laboratory **Laboratory classes** 1. **Introduction to Threads** Create basic threads; simulate parallel motion of particles in free fall. 2. **Parallel Harmonic Oscillators** Run multiple oscillators in parallel and compare execution times.

	3. **Shared Data Synchronization** Simulate radioactive decay with safe updates of a shared counter. 4. **Parallel Speedup Tests** Monte Carlo integration with many threads; analyze speedup. 5. **Locks and Conditions** Model random walkers with coordinated access to shared data. 6. **Producer Consumer Pattern** Simulate a detector where events are generated and analyzed concurrently. 7. **Parallel Histogramming** Construct particle energy histograms using thread safe structures. 8. **Readers and Writers** Concurrent access to a shared set of measurement results. 9. **Deadlock Scenarios** Demonstrate deadlock in competing processes and apply avoidance methods. 10. **Executors and Futures** Parallel random walk simulations with collection of statistics. 11. **Barrier Synchronization** Synchronize oscillators at fixed time steps to model collective motion. 12. **Fork Join Algorithms** Parallel summation of large datasets from physical simulations. 13. **Parallel Stream Processing** Compute and reduce particle energies in parallel. 14. **Parallel Diffusion Model** Simulate two dimensional diffusion and study performance scaling. 15. **Integrated Simulation Project** Combine decay simulation and histogramming; compare performance.
Prerequisites and co-requisites	
Assessment methods and criteria	see table below

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	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. 2. T. Rauber, G. Rünger, Parallel Programming: for Multicore and Cluster Systems, Springer Nature, Berlin, 2010. 3. S. Selikoff, J. Boyarsky, OCA/OCP Java SE 8 programmer: practice tests, Sybex, Indianapolis, Indiana, 2017.

	Supplementary literature	1. M. Ben-Ari, "Principles of Concurrent and Distributed Programming", 2nd edition, Addison Wesley, Upper Saddle River, NJ, 2006. 2. R.-G. Urma, M. Fusco, A. Mycroft, Modern Java in Action, Manning Publications, Shelter Island, 2018. 3. B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, D. Lea, Java Concurrency in Practice. Addison-Wesley, Upper Saddle River, NJ, 2006. 4. B. J. Evans, J. Clark, M. Verburg, The Well-Grounded Java Developer, Second Edition, Manning Publications, Shelter Island, 2023
	eResources addresses	
Example issues/ example questions/ tasks being completed	1. Calculate the Speedup of a concurrent program. 2. Convert a sequential program in a concurrent program.	
Practical activites within the subject	Not applicable	

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