AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Module name: The Art of Multiprocessor Programming

Academic year: 2019/2020 Code: ZSDA-3-0090-s ECTS credits: 3

Faculty of: Szkoła Doktorska AGH

Field of study: Szkoła Doktorska AGH Specialty: —

Study level: Third-cycle studies Form and type of study: Full-time studies

Lecture language: English Profile of education: Academic (A) Semester: 0

Course homepage: —

Responsible teacher: Kułakowski Konrad (kkulak@agh.edu.pl)

Module summary

The aim of the lecture is to deepen and consolidate knowledge about concurrent programming in Java programming language.

Description of learning outcomes for module

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Connections with FLO	Method of learning outcomes verification (form of completion)
Social competence: is able to			
M_K001	student is able to critically evaluate scientific achievements in the field of operating systems	SDA3A_K01	Activity during classes
M_K002	student is able to critically evaluate scientific achievements in the field of parallel programming	SDA3A_K01	Activity during classes
Skills: he can			
M_U001	the student is able to create parallel and concurrent software solutions	SDA3A_U01	Activity during classes
Knowledge: he knows and understands			
M_W001	The student understands the basics of multiprocessor programming	SDA3A_W01	Activity during classes

Number of hours for each form of classes

	Form of classes										
Suma	Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Prace kontrolne i przejściowe	Lektorat
30	14	0	0	0	0	16	0	0	0	0	0

FLO matrix in relation to forms of classes

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Form of classes										
		Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Prace kontrolne i przejściowe	Lektorat
Social competence: is able to												
M_K001	student is able to critically evaluate scientific achievements in the field of operating systems	+	-	-	-	-	+	-	-	-	-	-
M_K002	student is able to critically evaluate scientific achievements in the field of parallel programming	+	-	-	-	-	+	-	-	-	-	-
Skills: he can												
M_U001	the student is able to create parallel and concurrent software solutions	+	-	-	-	-	+	-	-	-	-	-
Knowledge: he knows and understands												
M_W001	The student understands the basics of multiprocessor programming	+	-	-	-	-	+	-	-	-	-	-

Student workload (ECTS credits balance)

Student activity form	Student workload
Udział w zajęciach dydaktycznych/praktyka	30 h
Preparation for classes	15 h
przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	30 h
Realization of independently performed tasks	15 h
Summary student workload	90 h
Module ECTS credits	3 ECTS

Additional information

Module content

Lectures

1) Introduction – shared objects, synchronization – basic synchronization models in Java
2) Mutual exclusion
3) Principles of concurrent objects I
3) Principles of concurrent objects II
4) Foundation of shared memory
5) The Relative Power of Primitive Synchronization Operations
6) Universality of consensus

Seminar classes

Sample topics covered by the seminary

1) Spin Locks and Contention
2) Monitors and Blocking Synchronization
3) Linked Lists: The Role of Locking
4) Concurrent Queues and the ABA Problem
5) Concurrent Stacks and Elimination
6) Counting, Sorting, and Distributed Coordination
7) Concurrent Hashing and Natural Parallelism
8) Skiplists and Balanced Search
9) Priority Queues
10) Futures, Scheduling, and Work Distribution
11) Concurrent Barriers
12) Transactional Memory

Teaching methods and techniques:

Lectures: lecture, presentation
Seminar classes: seminary, presentation

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu:

presence at the lectures and seminaries, presentation at the seminary

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa:

Lectures:
– Attendance is mandatory: No
– Participation rules in classes: compulsory attendance at the announced number of lectures
Seminar classes:
– Attendance is mandatory: Yes
– Participation rules in classes: compulsory attendance at the announced number of seminaries (usually all but one).

Method of calculating the final grade
based on presence and presentation at the seminary

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:
colloquium based on the given materials, books, articles + presentation

Prerequisites and additional requirements
good knowledge of the basics of Java programming language

Recommended literature and teaching resources
Herlihy, Shavit, The Art of Multiprocessor Programming, Elsevier, 2012

Scientific publications of module course instructors related to the topic of the module
- K. Kułakowski (2014) A concurrent van Emde Boas array as a fast and simple concurrent dynamic set alternative. Concurrency and Computation: Practice and Experience 26 (2) pp. 360–379.
- K. Kułakowski (2012) Concurrent systems modeling with CCL. Automatyka
- K. Kułakowski, J. Wąs (2010) Two Concurrent Algorithms of Discrete Potential Field Construction. In Parallel Processing and Applied Mathematics. pp. 529-538. Springer.

Additional information
None