NAME OF THE COURSE	Parallel Programming					
Code	PMID40	Year of study				
Course teacher	prof.dr. sc. Marko Rosić dr. sc. Tonći Dadić	Credits (ECTS)	5,0			
Associate teachers	Marin Aglić Čuvić mag. educ. inf.	Type of instruction (number of hours)	L	S	E	F
			30		30	
Status of the course		Percentage of application of e-learning				

COURSE DESCRIPTION	
Course objectives	Understanding the parallel execution of the program and the acquisition of knowledge and skills to achieve the program based on parallel algorithms.
Course enrolment requirements and entry competences required for the course	Object oriented programming.
Learning outcomes expected at the level of the course (4 to 10 learning outcomes)	Students will be able to: 1. Explain models of parallel program execution. 2. Understand and explain the terms of the process, nor (Eng. Thread), bidding either to access shared data, critical sections, synchronization and even deadlock. 3rd Apply Amdahl's law to assess the acceleration parallel execution of a given program. 4th Self build some simple parallel algorithms. 5 Understand some advanced parallel algorithms and apply them in the given problems. 6th implement and evaluate parallel programs.
Course content broken down in detail by weekly class schedule (syllabus)	Week 1: Parallel programming basics Why parallel programming? Moore's Law and multicore processors Simultaneous execution of the program The objectives of parallelization Evaluation criteria parallel algorithm Amdalov law acceleration of parallel programs Week 2: Parallel programming basics (continued) Parallelism, communication and coordination program Program structure for the coordination of simultaneous programs Software errors specific to parallel programs Competition for access to shared data (competitive read / write and write / write) The lack of progress of the program: a complete failure and even starvation Week 3: Parallel computer architecture multicore processors Shared and distributed memory Architecture and SIMD vector processing Architecture MIMD Week 4: Parallel Computer Architecture (continued) Terminology by Flynn

	Model synchronous PRAM computer Model asynchronous PRAM computer Processor instructions indivisible cycle of reading and writing of memory Week 5: Parallel algorithms, analysis and programming Acceleration and scalability Natural parallel algorithms A parallel approach: divide and conquer, reduciraj, leader-companions Week 6: Parallel algorithms, analysis and programming (continued) Some specific algorithms: Merge and Quick varieties Parallel algorithms for search graph Parallel matrix operations Producer - consumer Week 7: Reduction algorithm for an arbitrary number of processors Algorithm sum of prefixes for an arbitrary number of processors Algorithm reduction for a limited number of processors Algorithm sum of prefixes for a limited number of processors Week 8: Communication and coordination Changing data in a strongly connected parallel system Data exchange in the loosely connected system Week 9: Standard: MPI (Eng. Message Passing Interface) Individual and collective messaging Blocking and non-blocking messaging The role of the order sending or receiving messages Atomicity Week 10: Communication and coordination (continued) Specification and testing Atomicity and safety requirements Grain atomic data access and transactions Mutual exclusion even with the help of locking semfora and monitor Necessary conditions occurrence of deadlock and its prevention Transactions: optimistic and pessimistic approach Week 11: parallel decomposition Interference nor the concept of the critical section The need for communication and coordination and synchronization of threads Synchronization using a traffic light and active waiting The division of tasks partitioning of common data Week 12: Parallel decomposition (continued) Interference nor the concept of the critical section The need for communication and coordination and synchronization of threads Synchronization using a traffic light and active waiting The division of tasks partitioning of common data Week 13: Parallel decomposition (continued) Basic concepts of parallel decomposition Decomposition based on tasks The implementation of parallelism using either (Eng. Threads) Strategy SIMD

	Week 14: Evaluation of parallel programs Measurement time characteristics of the program load balancing Week 15: Evaluation of parallel programs (continued) Determining the time of communication between threads / processes Parallel queries the database Performance caching at the time of execution of the program
Format of instruction	☒ lectures ☐ seminars and workshops ☒ exercises ☐ on line in entirety ☐ partial e-learning ☐ field work ☐ independent assignments ☐ multimedia ☐ laboratory ☐ work with mentor ☒ homework assignments
Student responsibilities	attendance 70% lectures and 70% exercises and two homework.

Screening student work *(name the proportion of ECTS credits for each activity so that the total number of ECTS credits is equal to the ECTS value of the course)*

Name	Ects	Name	Ects	Name	Ects
Class attendance	0.5	Research		Experimental work	
Oral exam	1	Report		Homework assignments	0.5
Seminar essay		Essay			
Tests		Practical training			
Written exam	3	Project			

Grading and evaluating student work in class and at the final exam	Attendance 10%, homeworks 10%, practical / written examination or colloquium 40% and oral exam 40%.

Required literature (available in the library and via other media)

Title	Number of copies in the library	Availability via other media
Domagoj Jakobović: "Lectures from the course Parallel Programming", FER, Zagreb, 30.03.2015.	0	
http://www.fer.unizg.hr/_download/repository/Paralelno_programiranje_predavanja%5B8%5D.pdf (available 06/10/2015)	0	

Optional literature (at the time of submission of study programme proposal)	
Quality assurance methods that ensure the	Talk with students, student evaluation using the anonymous survey, the success of students in the exam, self-assessment.

acquisition of exit competences	
Other (as the proposer wishes to add)	