Course title	Code	semester	T+U	credit	ECTS
Parallel Programming		3	3+0	3	4

Prerequisite Courses	None
Language of the Course	English
Course Level	Undergraduate
Type of Course	Optional
Course Coordinator	
Instructors	
Course Assistants	
The aim of lesson	To explain the theory of parallel computers and programming and parallel systems to develop advanced software for
Course Content	Classification of parallel computer systems, levels of parallelism, parallel operations. Petri nets; of parallel organizations across parallel processes definition and coordination. Parallel processing concept. Basic parallelism; SISD computers and multiprocess CPUs. Pipeline computers; MISD computers, linear and nonlinear; super scalar and super pipeline computers. Asynchronous parallelism. MIMD systems. MIMD programming languages and coarse grain parallel algorithms. Synchronous parallelism. Structure of SIMD systems. Communication in SIMD systems. SIMD programming languages and MasPar algorithms. Non-procedural parallel programming languages.
Course Learning Outcomes	At the end of this course, the student; 1. I can give the development of parallel computers and how modern parallel computers are developed. can explain how it works. 2. Will be able to classify parallel computers and parallel computation models. 3. Parallel applications implemented on different parallel computers can be evaluated and compared with each other.

Weeks	Topics
one	Parallel Computers
2	Classification of Parallel Computer Systems, Levels of Parallelism, Parallel Operations
3	Petri Nets; Identifying Parallel Organizations Among Parallel Processes and coordination
4	Parallel Computers and Network Structures
5	Basic Parallelism; SISD Computers and Multiprocess CPUs
6	Pipeline Computers; MISD Computers.
7	Asynchronous Parallelism
8	Structure of MIMD Systems. Synchronization and Communication in MIMD Systems
9	MIMD Programming Languages and Coarse Grain Parallel Algorithms
10	Synchronous Parallelism
11th	Structure of SIMD Systems, Communication in SIMD Systems and Quiz
12	SIMD Programming Languages and Maspar Algorithms
13	Perception of Parallelism; Automatic Parallelization
14	Perception of Parallelism; Vectorization

General Competencies

Writes code using parallel programming technique.
Develops applications using multiple cores.
Evaluates performance by understanding the differences between serial programming and parallel programming.

resources
Braunl, T., (1993). Parallel Programming an introduction, Prentice Hall. Hwang, K., (1993). Advanced computer architecture; parallelism, scalability and programmability, McGraw Hill.
Evaluation System
The dates, days and hours of the Midterm Exam, Quiz, Final Exam and Evaluations will be announced later, according to the decision of the Faculty Administrative Board.

	WITH PROGRAM LEARNING OUTCOMES **COURSE LEARNING OUTCOMES RELATIONSHIP TABLE**										
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
LO1	3	2	one	3	2	2	one	2	2	one	one
LO2	3	3	2	3	2	one	2	2	3	one	2
LO3	3	3	2	3	2	one	one	one	2	2	one
LO: Learning Outcomes OP: Program Outcomes											
Contribution Level	**1 Very Low**		**2 Low**		**3 Medium**		**4 High**		**5 Very High**		

Relation of Program Outcomes and Related Course

Lesson	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**
Parallel Programming with CPU	3	3	2	3	2	one	one	2	2	one	one