COURSE OUTLINE

(1) GENERAL

SCHOOL	Engineering			
DEPARTMENT	Electrical and Computer Engineering			
LEVEL OF STUDY	Undergraduate			
COURSE UNIT CODE	8.028	**SEMESTER**	8^{th}	
COURSE TITLE	Parallel Processing			
	COURSEWORK BREAKDOWN		**TEACHING WEEKLY HOURS**	**ECTS Credits**
	Theory (Lectures)		2	2
	Practice		2	1
	Lab		1	1
	TOTAL		**5**	**4**
COURSE UNIT TYPE	Specialized general knowledge/Skills development			
PREREQUISITES	None			
LANGUAGE OF INSTRUCTION/EXAMS	Greek			
COURSE DELIVERED TO ERASMUS STUDENTS	Yes			
WEB PAGE (URL)	https://eclass.hmu.gr/courses/			

(2) LEARNING OUTCOMES

Learning Outcomes

The knowledge which students will acquire upon successful completion of the course relates to a deep understanding of what is a parallel computer and a parallel program. The students will examine various parallel computer architectures that have been established over the years and will understand their main design principles and functionality. In the second part of the course they will be exposed to parallel programming, understanding how to program a parallel machine, shared or distributed memory, using standard parallel programming models, such as OpenMP and the MPI message passing interface. The students will thsu acquire advanced programming skills in advanced programing environments.

To summarize, upon successful completion of the course, students will be able to:

- Understand the design principles of various parallel architectures.
- Understand the difference between different parallel programming models.
- Be able to design a parallel computational solution for a problem.
- Be able to implement a parallel algorithm with a parallel programming model and test its efficiency in a cluster of workstations environment.

General Skills

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adapt solutions to new situations
- Autonomous work
- Teamwork
- Decision making
- Work in an interdisciplinary environment

(3) SYLLABUS

Course description

The course aims to serve as an introduction to the main programming techniques for parallel computing systems. The most important architectural categories of parallel computers are covered, such as shared memory parallel computers, distributed memory parallel interconnection systems, multicore and manycore systems, clusters, etc. The programming part includes both the shared address space memory model and the message passing model, by examining popular standards such as OpenMP and MPI. In the last part of the course, specific parallel programming algorithms are examined, such as examples from the scientific computing area (computation of π), problems of data management and organization (i.e. sorting), optimization and load balancing problems. More specifically, the course will cover the following topics:

- **Introduction:** Parallel computers. Parallel computer architectures. Parallel programming. Main parallel programming models. Programming methodology. How can we parallelize a problem. Problem parallelization and dividing a problem into tasks.
- **Shared memory architectures:** Basic organization. Switching models. Multilevel interconnection networks. Shared memory and consistency. Main programming protocols.
- **Distributed memory architectures:** Basic organization. Main interconnection topologies. Routing protocols. Basic distributed memory programming models.
- **Message passing programming:** Basic structures. Programming using message passing. MPI programming. MPI collective communication primitives. Other MPI instructions. Parallel programming examples. Complexity of problems.

Registered students will be required to complete programming exercises in each of the studies models.

(4) TEACHING METHODS - ASSESSMENT

MODE OF DELIVERY	In-Class Face-to-Face
USE OF INFORMATION AND COMMUNICATION TECHNOLOGY	▪ Use of ICTs in lecturing and lab ▪ Use of ICTs for the communication with students via the e-class platform
TEACHING ORGANIZATION	(see table below)
ASSESSMENT METHODS	Language for Evaluation: Greek/English (Erasmus) All announcements related to the syllabus, including grading, and complementary reading material are permanently posted in the course web page (ECLASS). The course grade incorporates the following evaluation procedures: 1. Final exam (60%) 2. Midterm Exam (20%) 3. Programming exercises (20%)

Method description/Activity	Semester Workload
Lectures	40
Demos/Labs	30
Individual Study & Programming (on Server or Laptop/PC using VM or Dual Boot)	30
Total Contact Hours	**100**

(5) RECOMMENDED BIBLIOGRAPHY

- Recommended Bibliography:

- *Vasileios V. Dimakopoulos, "Parallel Systems and Programming", ISBN: 978-960-603-369-8, 1st revised edition 2017, Copyright SEAB, 2015, https://repository.kallipos.gr/bitstream/11419/3209/4/book.pdf (in Greek)*
- *Athanasios I. Margaris, "MPI Theory & Applications", ISBN: 978-960-418-145-2, Copyright 2019, 2014 Tziola Publications. (in Greek)*

- Additional Reading:

- *V. Kumar, A. Grama, A. Gupta, and G. Karypis, "Introduction to Parallel Computing: Design and Analysis of Algorithms", Benjamin Cummings Publishing Company, 1994.*
- *I. Foster, "Design and Building of Parallel Programs", Online book available from http://www.mcs.anl.gov/~itf/dbpp*
- *N. Matloff, "Programming on Parallel Machines: GPU, Multicore, Clusters and More", Online book, available from http://heather.cs.ucdavis.edu/parprocbook*