INSTITUTION	NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS					
SCHOOL	SCHOOL OF SCIENCE					
DEPARTMENT	INFORMATICS AND TELECOMMUNICATIONS					
COURSE LEVEL	UNDERGRADUATE					
COURSE TITLE	**Parallel Systems**					
COURSE CODE	**ΘΠ04**		**Semester**	**5**	**ECTS**	6
TEACHING HOURS per week	**THEORY**	3	**SEMINAR.**	1	**LABORATORY**	
COURSE TYPE	**Select one of the following and delete the rest** Elective (ΠΜ) **K**: A-B; **E1**: ; **E2**: E; **E3**: B; **E4**: B; **E5**: ; **E6**: *Fill the table as in the curriculum:* Track (A-Computer Science, B- Computer Engineering) / Specialization Compulsory (Y) / Core Specialization (B)/ Elective Specialization (E)					
URL	https://eclass.uoa.gr/courses/D36/					
EXPECTED PRIOR KNOWLEDGE/ PREREQUISITES AND PREPARATION:	Recommended K14 Architecture I					
TEACHING AND EXAMINATIONS LANGUAGE:	GREEK					
THE COURSE IS OFFERED TO ERASMUS STUDENTS	NO					

COURSE CONTENT
Covers the spectrum parallel architectures, parallel APIs, parallel application development and evaluation. Presents Flynn Classification and elaboration on MIMD architectures shared (UMA, NUMA, cc-NUMA) and distributed memory. GPUs as HPC. Measures of speed-up, efficiency, cost, Amdahl and Gustafson Laws. Foster design methodology, data and domain parallelism. Parallel programming APIs MPI, openMp, CUDA and hybrid combinations. Parallel Programming techniques for HPC applications. Profiling tools for evaluating and improving performance.

STUDENT LEARNING OBJECTIVES

Upon successful completion of the course the student will be able to:

- Classify architectures according toFlynn. Identify MIMD shared memory (UMA, NUMA, cc-NUMA) and distributed memory architectures.
- Define measures of speed-up, efficiency, cost and scalability as criteria of performance.
- Explain and apply Amdahl and Gustafson laws.
- Design, develop and evaluate parallel applications in MPI, OpenMp, CUDA and their hybrid combinations.
- Execute and debug parallel applications on HPC clusters
- Use profiling tools for improving application performance.

TEACHING AND LEARNING METHODS - ASSESSMENT	
TEACHING METHOD	In Class (Face to Face)
USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES	Learning process supported by the e-class platform (specify which specific tools you use, eg Teaching material; Discussions on project, laboratories, etc; Announcements; Task assignments; Student groups) Email communication Live transmission of lectures Ability to track recorded lectures Programming Seminars
TEACHING ORGANIZATION *Describe in detail the way and methods of teaching:* *Enhanced Lectures,* *Online Lectures,* *Seminars,* *Tutorial,* *Laboratory,* *Laboratory Exercise,* *Study & analysis of literature,* *Practice (Positioning),* *Interactive teaching,* *Developing a project,* *Individual / group work* *Telework (reference to tools) etc.* *Details of the student's study hours for each learning activity and hours of non-guided study are shown to ensure that the total workload at the semester corresponds to the ECTS*	

Activity	***Student Workload*** *(hours)*
Lectures	39
Seminars	13
Study MPI Laboratory	15
Study OpenMp Laboratory	10
Study CUDA Laboratory	10
Tools	8
Programming Assignment	35
Independent Study	20
Total Course (25 hours of workload per unit of credit)	***150***

ASSESSMENT OF STUDENTS *Description of the assessment process* *Assessment Methods, Formative or Concluding, Multiple Choice Test, Quick Response Questions, Test Development Questions, Problem Solving, Written Work, Report / Report, Oral Examination, Public Presentation, Laboratory Work, Other / Other* *Fully defined evaluation criteria are mentioned and if and where they are accessible to students.*	Evaluation of theory by written examination and programming by a project (teams of 2-3). Grade Feedback is given in the written examination at the level of questions. In the project a rubric is given and feedback is provided accordingly with explanations in case of missing grades.

Assessment methods	***Number***	***Percentage***
Written examination	1	30%
Final work	1	70%

LITERATURE AND STUDY MATERIALS / READING LIST
Basic Book Peter S. Pacheco, Introduction to Parallel Programming (Greek edition) Supportive Book Drimakopoulos V parallel systems and programming (on-line) http://hdl.handle.net/11419/3209 Notes, presentaions, tutorials on Parallel programming and tools