

COURSE OUTLINE

1. GENERAL

SCHOOL		SCHOOL OF ENGINEERING, UNIVERSITY OF PATRAS	
DEPARTMENT		DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATICS	
LEVEL OF COURSE		UNDERGRADUATE, ELECTIVE	
COURSE CODE		CEID_NE5407	SEMESTER OF STUDIES WINTER
COURSE TITLE		SOFTWARE AND PROGRAMMING OF HIGH PERFORMANCE SYSTEMS	
INDEPENDENT TEACHING ACTIVITIES <i>if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits</i>		TEACHING HOURS PER WEEK	ECTS CREDITS
Lectures		2	2
Recitation Sections		1	1
Laboratory Exercises		2	2
TOTAL		5	5
<i>Add rows if necessary. The teaching organization and methods used are described in detail in (d).</i>			
COURSE TYPE <i>general background, special background, specialized general knowledge, skills development</i>		Skills development	
PREREQUISITE COURSES:		The course requires knowledge acquired from the following courses: «Parallel Processing», «Operating Systems» and «Computer Architecture».	
TEACHING AND ASSESSMENT LANGUAGE:		Greek. It is possible to be delivered in English if there are foreign students.	
THE COURSE IS OFFERED TO ERASMUS STUDENTS		YES	
COURSE WEBPAGE (URL)		https://eclass.upatras.gr/courses/CEID1246/	

2. LEARNING OUTCOMES

Learning outcomes <i>The course learning outcomes, specific knowledge, skills, and competences of an appropriate level that the students will acquire with the successful completion of the course are described.</i> <i>Consult Appendix A</i> • <i>Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area</i> • <i>Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B</i> • <i>Guidelines for writing Learning Outcomes</i>
At the end of this course, the student will: • understand the transition from serial to parallel computing, the connection between the different categories of parallel systems, and the techniques for faster application execution and data processing. • know how to apply code optimization techniques to achieve better utilization of the underlying hardware and exploit performance metrics for a deeper understanding of application performance. • be familiar with utilizing parallelism in applications and programming environments in the field of machine learning. • know ways to utilize task parallelism in basic programming languages (C, Python) and techniques for efficient data processing and parameter optimization in machine learning. Moreover, the student will be able to: • exploit the tools available at the system, software, and algorithm levels for application development in the broader area of Computational Science. • program shared and distributed memory computing systems using programming models such as OpenMP and MPI. • Program computing systems with graphics cards using programming models such as CUDA and OpenACC.

General Abilities <i>Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?</i>	
Search for, analysis and synthesis of data and information, with the use of the necessary technology situations	Project planning and management Respect for difference and multiculturalism Respect for the natural environment
Decision-making	Showing social, professional and ethical responsibility and sensitivity to gender issues
Working independently	Criticism and self-criticism
Team work	Production of free, creative and inductive thinking
Working in an international environment	Working in an interdisciplinary environment
Production of new research ideas	Others...

Search for, analysis and synthesis of data and information with the use of the necessary technology
Adapting to new situations
Decision-making
Team work
Production of free, creative, and inductive thinking

3. COURSE CONTENT

Categories of parallel systems. Parallelization of code in shared and distributed memory systems. The OpenMP and MPI parallel programming models. Management of large volumes of data with parallel I/O techniques. Code optimization techniques and exploitation of performance metrics. Basic features of graphics accelerators and their exploitation with the CUDA and OpenACC parallel programming models. Exploitation of parallelism in applications with an emphasis on stochastic optimization algorithms, machine learning problems, and large-scale neural network training. Exploitation of task parallelism in the Python programming language. Techniques for efficient data processing and parameter optimization in machine learning.
--

4. TEACHING AND LEARNING METHODS - ASSESSMENT

TEACHING METHOD <i>Face-to-face, Distance learning, etc.</i>	Face-to-face												
USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES <i>Use of ICT in teaching, laboratory education, communication with students</i>	We use Information and Communications Technology in communication with students. We use e_class, e_mail.												
TEACHING ORGANIZATION <i>The manner and methods of teaching are described in detail. Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc. The student's study hours for each learning activity are given, as well as the hours of non-directed study according to the principles of the ECTS</i>	<table> <tr> <th>Activity</th><th>Semester Workload</th></tr> <tr> <td>Lectures</td><td>26</td></tr> <tr> <td>Recitation</td><td>13</td></tr> <tr> <td>Laboratory Exercises</td><td>52</td></tr> <tr> <td>Study</td><td>39</td></tr> <tr> <td>Total number of hours for the Course</td><td>130</td></tr> </table>	Activity	Semester Workload	Lectures	26	Recitation	13	Laboratory Exercises	52	Study	39	Total number of hours for the Course	130
Activity	Semester Workload												
Lectures	26												
Recitation	13												
Laboratory Exercises	52												
Study	39												
Total number of hours for the Course	130												
STUDENT ASSESSMENT <i>Description of the evaluation procedure Language of evaluation, methods of evaluation, summative or conclusive, multiple-choice questionnaires, short-answer questions, open-ended questions, problem-solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other Specifically defined evaluation criteria are given and if and where they are accessible to students.</i>	• Language of evaluation: Greek (English if required) • Written examination (30% of the final grade) • Laboratory exercises (70% of the final grade)												

5. RECOMMENDED LITERATURE

Suggested bibliography:

- Course slides (available through the e-class platform, English).
- Programming Massively Parallel Processors, 2nd Edition. David B. Kirk, Wen-mei W. Hu, Morgan-Kaufmann, 2012.

Related academic journals:

- IEEE Transactions on Parallel and Distributed Systems
- ACM Transactions on Parallel Computing
- International Journal of Parallel Programming
- Journal of Parallel and Distributed Computing
- Parallel Computing