COURSE OUTLINE

(1) GENERAL

SCHOOL	SCHOOL OF ENGINEERING		
ACADEMIC UNIT	COMPUTER ENGINEERING AND INFORMATICS		
LEVEL OF STUDIES	UNDERGRADUATE		
COURSE CODE	CEID_NY408	SEMESTER	SPRING
COURSE TITLE	Parallel Processing		

INDEPENDENT TEACHING ACTIVITIES *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*	WEEKLY TEACHING HOURS	CREDITS
Lectures, Recitation sections, Laboratory Exercises	2(L), 1(RS), 3(LE)	6
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		6

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Skills development
PREREQUISITE COURSES:	The course requires knowledge acquired from the following courses: “Introduction to Computers & Programming”, “Introduction to Algorithms”, “Topics in Computer Architecture”, “Operating Systems”
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek. It is possible to be delivered in English if there are foreign students.
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes (in English)
COURSE WEBSITE (URL)	https://eclass.upatras.gr/courses/CEID1057/

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

After attending the course the students should be able to:

1. Identify the different categories of Parallel Systems and evaluate their advantages and disadvantages
2. Evaluate the performance of a parallel application using appropriate performance metrics
3. Describe what is a critical area, mutual exclusion, semaphore and atomic instructions
4. Identify when mutual exclusion is required in parts of a parallel application
5. Identify on which parallel architecture a particular programming model can be used
6. Develop a parallel application using any of the POSIX Threads, OpenMP and MPI models
7. Differentiate among types of data dependencies
8. Explain what vectorization is and what its benefits are
9. Explain the purpose of interconnection networks
10. Differentiate among categories of interconnection networks
11. Explain how some of the interconnection networks operate

After attending the course the students will have developed the following skills:

1. They will be able to identify the parts of a serial application that have increased chances of

achieving better performance when they are parallelized
2. They will be able to develop a parallel application using the appropriate tools and programming models for the computing system used
3. They will be able to evaluate the performance of the parallel application they have created
4. They will be able to identify the parts of the parallel application that can be optimized
5. They will be able to apply techniques to optimize performance at these points

General Competences

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?

Search for, analysis and synthesis of data and information, with the use of the necessary technology
Adapting to new situations
Decision-making
Working independently
Team work
Working in an international environment
Working in an interdisciplinary environment
Production of new research ideas
Project planning and management
Respect for difference and multiculturalism
Respect for the natural environment
Showing social, professional and ethical responsibility and sensitivity to gender issues
Criticism and self-criticism
Production of free, creative and inductive thinking
……
Others…
…….

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

(3) SYLLABUS

1. Introduction to the concepts of Parallel Processing
2. Classification of Parallel Computing Systems
   a. Flynn's taxonomy
   b. Classification based on the architecture of the memory
   c. Vector architectures
   d. Systolic arrays
3. Performance metrics
   a. Amdahl's Law
   b. Gustafson's Law
   c. Speedup
   d. Karp-Flatt metric
4. Mutual exclusion - Semaphores - Atomic instructions
5. Programming models for parallel architectures
   a. Threads - The POSIX Threads programming model
   b. The OpenMP programming model
      i. Execution model
      ii. Directives
      iii. Functions
      iv. Environment variables
      v. Tasks - User-level scheduling
   c. The MPI programming model
      i. Execution model
      ii. Point-to-point communication
      iii. Collective communication
      iv. Topologies
6. Data dependencies - Vectorization
7. Interconnection networks
   a. General concepts
   b. Static interconnection networks
      i. One-dimensional, two-dimensional, Hypercube
   c. Dynamic interconnection networks

i. Bus
ii. Single-stage
iii. Multi-stage – Shuffles
iv. Crossbar

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Face-to-face
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	1. The slides and any other supporting material is available through the e-Class platform of the course. 2. All lectures of the course are digitally available through the OPEN COURSES platform.
TEACHING METHODS *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*	

Activity	*Semester workload*
Lectures	2*13=26
Recitation sections	1*13=13
Laboratory Exercises	3*13=39
Study and preparation with the help of the OPEN COURSES platform that contains video-conferencing lectures	3*13=39
Weekend study	2*13=26
Final exam preparation week+2 weeks break	4*3=12
Course total (~26 hours workload per credit point)	***155***

STUDENT PERFORMANCE EVALUATION *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.*	1. Language of evaluation: Greek (English if required) 2. Final examination (70% of final mark): Written exam that might include multiple choice questionnaires, short-answer questions, problem solving 3. Laboratory exercise (30% of final mark): The exercise requires the application of parallel programming techniques, optimization, performance analysis and application of performance metrics. 4. Grading scale: 1 to 10 Minimum passing mark: 5 Grades < 5 correspond to ECTS grade F. For the rest of the grades the correspondence is as follows: >= 9: ECTS A >= 8: ECTS B >= 7: ECTS C >= 6: ECTS D >= 5: ECTS E

(5) ATTACHED BIBLIOGRAPHY

1. Suggested bibliography:
   - Course slides (available through the e-class platform, Greek).
   - An Introduction to Parallel Programming, 1st Edition. Peter S. Pacheco, Morgan-Kaufmann,

2011.

- Programming and Architecture of Parallel Processing Systems. Stylianos Papadakis, Konstantinos Diamantaras, 2012 (Greek).
- Parallel Systems and programming, Vasileios Dimakopoulos, 2017 (Greek).

2. 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