

## COURSE OUTLINE

### GENERAL

<b>SCHOOL</b>	Sciences and Engineering		
<b>ACADEMIC UNIT</b>	Computer Science		
<b>LEVEL OF STUDIES</b>	1 <sup>st</sup> Cycle		
<b>COURSE CODE</b>	COMP-417	<b>SEMESTER</b>	Fall
<b>COURSE TITLE</b>	Parallel and Distributed Computing		
<b>INDEPENDENT TEACHING ACTIVITIES</b> <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>		<b>WEEKLY TEACHING HOURS</b>	<b>CREDITS</b>
		2.5	6
<i>Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).</i>			
<b>COURSE TYPE</b> <i>general background, special background, specialised general knowledge, skills development</i>	Special Background		
<b>PREREQUISITE COURSES:</b>	COMP-212, COMP-354		
<b>LANGUAGE OF INSTRUCTION and EXAMINATIONS:</b>	English		
<b>IS THE COURSE OFFERED TO ERASMUS STUDENTS</b>			
<b>COURSE WEBSITE (URL)</b>			

### LEARNING OUTCOMES

<p><b>Learning outcomes</b></p> <p><i>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.</i></p> <p><i>Consult Appendix A</i></p> <ul style="list-style-type: none"> <li>• <i>Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area</i></li> <li>• <i>Descriptors for Levels 6, 7 &amp; 8 of the European Qualifications Framework for Lifelong Learning and Appendix B</i></li> <li>• <i>Guidelines for writing Learning Outcomes</i></li> </ul>
<p>After completion of the course students are expected to be able to:</p> <ul style="list-style-type: none"> <li>• critically evaluate the difference between shared-memory and distributed-memory architectures and corresponding parallel programming languages</li> <li>• apply Amdahl's law and Gustafson's law to derive an upper bound on the achievable speedup of a parallel program</li> <li>• critically evaluate the terms speedup, efficiency, scalability, and computation-to-communication ratio</li> <li>• analyze problems and apply applicable parallel and distributed solutions when developing software</li> </ul>

- correctly apply advanced concurrent concepts like threads, state guarding, atomicity, and locking for developing parallel applications
- compare and contrast the difference between implicit and explicit concurrency programming techniques
- critically evaluate current trends in parallel computing
- develop (write/debug/correct) parallel applications that satisfy safety and liveness properties in various programming languages.

### 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...*  
*.....*

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  
 Project planning and management  
 Criticism and self-criticism  
 Production of free, creative and inductive thinking

## SYLLABUS

1. Introduction Parallel and Distributed Computing
  - a) Shared memory vs distributed memory computing systems
  - b) Amdahl's law and Gustafson's law
  - c) Partitioning, communication, synchronization, and load balancing
2. Parallel Computing using Shared Memory
  - a) Processes and threads
  - b) Parallel program development using fork-join and divide-and-conquer patterns
  - c) Different optimization techniques
3. Shared Memory Concurrency and Mutual Exclusion
  - a) Concurrent Programming
  - b) Data race and bad interleaving
  - c) Consistent locking, coarse-grained vs fine-grained locking, critical-section granularity
  - d) Programming mechanisms for mutual exclusion
4. Process Synchronization

<ul style="list-style-type: none"> <li>a) Condition synchronization</li> <li>b) Programming mechanisms for process synchronization</li> <li>c) Nested locking issues</li> <li>d) Deadlock examples and analysis</li> </ul>
<ul style="list-style-type: none"> <li>5. Deadlock and Starvation <ul style="list-style-type: none"> <li>a) Liveness and safety properties</li> <li>b) Deadlock vs. Starvation</li> <li>c) Programming techniques to avoid deadlock and starvation</li> </ul> </li> <li>6. Implicit Parallel Computation using Shared Memory <ul style="list-style-type: none"> <li>a) Implicit vs explicit programming techniques</li> <li>b) Introduction to OpenMP</li> </ul> </li> <li>7. Current Trends in Parallel Computing <ul style="list-style-type: none"> <li>a) Current architectures used for supercomputers</li> <li>b) Current Frameworks/Middleware/Compilers used for developing parallel applications</li> </ul> </li> </ul>

## (1) TEACHING and LEARNING METHODS - EVALUATION

<b>DELIVERY</b> <i>Face-to-face, Distance learning, etc.</i>	Face-to-face																
<b>USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY</b> <i>Use of ICT in teaching, laboratory education, communication with students</i>	<i>Use of ICT in teaching / Χρήση ΤΠΕ</i> <i>Communication with students / Επικοινωνία με Φοιτητές</i>																
<b>TEACHING METHODS</b> <i>The manner and methods of teaching are described in detail.</i> <i>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.</i> <i>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><b>Activity</b></th><th><b>Semester workload</b></th></tr> <tr> <td>Lectures</td><td>35</td></tr> <tr> <td>Preparation, homework, quizzes</td><td>50</td></tr> <tr> <td>Project</td><td>15</td></tr> <tr> <td>Exam preparation</td><td>48</td></tr> <tr> <td>Final Exam</td><td>2</td></tr> <tr> <td></td><td></td></tr> <tr> <td>Course total</td><td><b>150</b></td></tr> </table>	<b>Activity</b>	<b>Semester workload</b>	Lectures	35	Preparation, homework, quizzes	50	Project	15	Exam preparation	48	Final Exam	2			Course total	<b>150</b>
<b>Activity</b>	<b>Semester workload</b>																
Lectures	35																
Preparation, homework, quizzes	50																
Project	15																
Exam preparation	48																
Final Exam	2																
Course total	<b>150</b>																
<b>STUDENT PERFORMANCE EVALUATION</b> <i>Description of the evaluation procedure</i> <i>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</i> <i>Specifically-defined evaluation criteria are given, and if and where they are accessible to students.</i>	Final Exam, Midterm Exam, Assignments, Project, and Quizzes																

## ATTACHED BIBLIOGRAPHY

### Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Parallel Programming: Concepts and Practice	Bertil Schmidt, Jorge Gonzalez-Dominguez, Christian Hundt, Moritz Schlarb	Morgan Kaufmann	2017	978-0128498903
Topics in Parallel and Distributed Computing	S. Prasad, A. Gupta, A. Rosenberg, A. Sussman, C. Weems	Elsevier	2015	978-0128039380

### Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
The Art of Multiprocessor Programming, 2nd Edition	Maurice Herlihy, Nir Shavit, Victor Luchangco, Michael Spear	Morgan Kaufmann	2020	978-0124159501
Java Concurrency in Practice	B. Goetz, T. Peierls, J. Bloch, J. Bowbeer, D. Holmes, D. Lea	Addison-Wesley Professional	2006	978-0321349606