## COURSE OUTLINE

## (1) GENERAL

| | | | |
|---|---|---|---|
| **SCHOOL** | Engineering | | |
| **DEPARTMENT** | Electrical and Computer Engineering | | |
| **LEVEL OF STUDY** | Undergraduate | | |
| **COURSE UNIT CODE** | 8.010 | **SEMESTER OF STUDY** | 8th |
| **COURSE TITLE** | Computer Architecture | | |

| **COURSEWORK BREAKDOWN** | **TEACHING WEEKLY HOURS** | **ECTS Credits** |
|---|---|---|
| Theory (Lectures) | 3 | 3 |
| Tutorial/Project | 1 | 1 |
| Laboratory | 1 | 1 |
| | | |
| **TOTAL** | 5 | 5 |

| | |
|---|---|
| **COURSE UNIT TYPE** | Specialized general knowledge/Skills development |
| **PREREQUISITES** | Computer Organization |
| **LANGUAGE OF INSTRUCTION/EXAMS** | Greek |
| **COURSE DELIVERED TO ERASMUS STUDENTS** | Yes |
| **WEB PAGE (URL)** | https://eclass.hmu.gr/courses/ |

## (2) LEARNING OUTCOMES

| **Learning Outcomes** |
|---|
| Computer Architecture is the continuation of Computer Organization and focuses on foundations of methods for design, evaluation and comparison of computing systems and in particular of the performance of the central microprocessor unit. At the same time the objective of this course is to provide students with a background on modern computer architecture trends. After completing this course, students should be able to:<br>• Recognize and analyze the performance of a computing system, such as the factors that impact the CPU execution time, memory performance<br>• Understand the basic concepts and methods of computer architecture including instruction-set architecture, pipelining, memory hierarchy, instruction execution parallelism, hardware threads<br>• Understand the modern methods for evaluation, optimization and comparison of computing systems<br>• Recognize and understand the cutting-edge technologies for design and implementation of multicore and manycore processors, of GPUs and of emerging memory technologies |
| **General Skills** |
| The objective of this course is for the student to acquire the following general skills:<br>• Quest, analysis, and composition of data and information through using the needed technologies<br>• Adapting to new situations<br>• Autonomous studying and collaborative studying<br>• Creative judgement, critical thinking<br>• Decision making<br>• Development of innovative research ideas |

## (3) SYLLABUS

This course focuses in techniques for quantitative analysis and evaluation of modern computing systems. It is based on the foundation course presenting the essential operation and organization of a computer. This course deals with modern methods for optimizing the performance and the energy consumption of a computing system. It will focus on pipelining techniques, parallelism at instruction level, virtual memory, memory hierarchies and multi-processors.

Fundamentals of Quantitative Design and Analysis – principles for design of computing systems, instruction-set architecture, cost and performance. Quantitative evaluation of computer performance through using benchmarks.

Memory Hierarchy Design – cache memory, parameters and effects on performance, virtual memory, address translation, protection, TLB.

Instruction Level Parallelism and Its Exploitation, Data-Level Parallelism in Vector, SIMD, and GPU Architectures, Thread-Level Parallelism – Parallelism techniques in hardware and software at the level of instructions, data and threads, branch prediction techniques and scheduling

Warehouse-Scale Computers to Exploit Request-Level and Data-Level Parallelism – Multicore processors, architectures for reduced power/energy, parallel memory systems, vector architectures, dataflow machines, and interconnection networks.

The laboratory focuses on analysis and assessment of quantitative performance through using simulators and modern tools.

## (4) TEACHING METHODS - ASSESSMENT

<table>
<tr><td>MODE OF DELIVERY</td><td>In-Class Face-to-Face</td></tr>
<tr><td>USE OF INFORMATION AND COMMUNICATION TECHNOLOGY</td><td><ul><li>Use of ICTs in lecturing and lab</li><li>Use of ICTs for the communication with students via the e-class platform</li></ul></td></tr>
<tr><td>TEACHING ORGANISATION</td><td><table>
<tr><th>Method description / Activity</th><th>Semester Workload</th></tr>
<tr><td>Lectures</td><td>50</td></tr>
<tr><td>Exercises</td><td>20</td></tr>
<tr><td>Labs</td><td>20</td></tr>
<tr><td>Individual Programming</td><td>20</td></tr>
<tr><td>Individual Study</td><td>40</td></tr>
<tr><td>Total Contact Hours</td><td>150</td></tr>
</table></td></tr>
<tr><td>ASSESSMENT METHODS</td><td>Language for Evaluation: Greek/English (Erasmus)<br><br>All announcements related to the syllabus, including grading, assessment criteria, and complementary reading material are permanently posted in the course web page (e-class). The course grade is based on the following evaluation procedures:<ol><li>Final exam on theoretical/practical problems (40%). The objective is to assess the students' understanding of key theoretical and practical concepts.</li><li>Intermediate exams (25%)</li><li>Individual take-home exercises/projects (15%)</li><li>Lab programming exercises (20%)</li></ol></td></tr>
</table>

## (5) RECOMMENDED BIBLIOGRAPHY


- *Textbook: Computer Architecture: A Quantitative Approach (5th Edition, Morgan Kaufmann, 2012), by John L. Hennessy and David A. Patterson*
- *Fundamentals of Parallel Multicore Architecture: Multichip and Multicore Systems, by Yan Solihin Publisher: CRC Press August 2016*

*Related Scientific Publications*
*Selected articles published in top conferences such as IEEE ISCA, MICRO, ASPLOS, HPCA, and journals (such as IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, IEEE Micro, ACM Transactions on Architecture and Code Optimization, ACM Transactions on Design Automation of Electronic Systems)*