COURSE OUTLINE

(1) GENERAL

SCHOOL	School of Sciences		
ACADEMIC UNIT	Department of Informatics		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	603	**SEMESTER**	6
COURSE TITLE	COMPUTER ARCHITECTURE		

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	2	5
Lab Exercises	2	
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Scientific Area, Skill Development
PREREQUISITE COURSES:	205
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek
IS THE COURSE OFFERED TO ERASMUS STUDENTS	No
COURSE WEBSITE (URL)	https://moodle.cs.duth.gr/course/view.php?id=22

(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*

Upon successful completion of the course, the student should:

1. Understands basic and critical computer architecture topics.
2. Has a specialized knowledge in advanced topics found in modern processors and computing systems.
3. Explains and solves problems related to modern processor cores and their instruction routing, both dynamic and static, or comments on topics from important publications.
4. Studies real processors, develops Assembly code, hardware description or high-level simulation with specialized tools.

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

- Search, analysis and synthesis of data and information, using the necessary technologies
- Project planning and management,
- Promotion of free creative and inductive thinking.

(3) SYLLABUS

1. Brief overview of the organization of a PC, with special emphasis on the microprocessor.

2. Performance evaluation of architecture based on metro programs.

3. Partial command overlap and control unit design to allow overlap.

4. Study of the possibility of overlapping execution units.

5. Multi-level overlay and hyperscale processors.

6. Dynamic instruction routing for out-of-order instruction execution based on watch table and commit stations, and implementation on modern processors.

7. Static instruction routing, very large instruction word (VLIW) processors, and special programming topics for static instruction routing.

8. Techniques for improving performance of modern processors with branch prediction models, hypothetical and assured execution.

9. Advanced memory and peripheral organization and access issues for high-performance processors.

10. An introduction to parallel architectures, from multiple flows of control to multiprocessors and multicomputers, as well as an introduction to cache coherency, memory coherency, and concurrency.

11. Develop skills in ARM RISC or RISCV processor Assembly language with appropriate software tools and in widely used hardware (Raspberry).

12. Develop skills in Edge Computing and architectural computing (labs).

(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*	Presentation with the help of slides, Course website with supporting and auxiliary material, Application of programming software in Assembly language, Contact by e-mail
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	26 h
Tutorial Exercises: Selected, representative exercises are solved concerning different modules of the course.	39 h
Written Exams	6 h
Exercises	28 h
Individual Study	26 h
Course total	125 h (5 ECTS)

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.*	Final written exam (60%) including theory questions and problem-solving topics. Delivery of individual or group semester study in the context of laboratory (40%). The final examination of students with dyslexia or other disabilities takes place after consultation.

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography:

Bibliography

1. «Computer Architecture: A Quantitative Approach» by John L. Hennessy, David A. Patterson, MK Publications.
2. «Digital Design and Computer Architecture» by Sarah Harris, David Harris, MK Publications.
3. «Αρχιτεκτονική Υπολογιστών», Δημήτριος Β. Νικολός, 2η Έκδοση.
4. «Συστήματα Υπολογιστών», Bryant – O'Hallaron, 3η Έκδοση, Κλειδάριθμος.

Journals

1. IEEE Transactions on Computers (TC)
2. ACM Transactions on Architecture and Code Optimization (TACO)
3. Future Generation of Computer Systems (FGCS- Elsevier)