COURSE DESCRIPTION

1. Program Information

1.1 University	**“Alexandru Ioan Cuza” University of Iaşi**
1.2 Faculty	**Computer Science**
1.3 Department	**Computer Science**
1.4 Study Domain	**Computer Science**
1.5 Study Cycle	**Bachelor**
1.6 Study Program / Qualification	**Computer Science**

2. Course Information

2.1 Course Name			**Computer Architecture and Operating Systems**				
2.2 Course Teacher			**Prof. dr. Henri Luchian, Lect. dr. Vlad Rădulescu**				
2.3 Seminary Teacher			**Lect. dr. Vlad Rădulescu**				
2.4 Study Year	**I**	**2.5** Semester	**1**	**2.6** Evaluation	**E**	**2.7** Course Status*	**OB**

** OB – Compulsory / OP – Optional*

3. Total estimated hours (hours per semester and didactic activities)

3.1 Hours per week	**4**	in which: **3.2** course	**2**	**3.3** seminary/laboratory	**2**
3.4 Hours in curriculum	**56**	in which: **3.5** course	**28**	**3.6** seminary/laboratory	**28**

Time Distribution	hours
Manual study, Course support, Bibliography, and others	**15**
Supplementary Documentation in library, in electronic forums, and on the field	**15**
Seminaries/laboratories preparation, homeworks, reports, portfolios and essays	**15**
Tutoring	**-**
Evaluation	**4**
Other activities (consultations per student)	**10**

3.7 Total hours individual study	**45**
3.8 Total hours per semester	**115**
3.9 Credits	**5**

4. Preconditions (if necessary)

4.1 Of Curriculum	**-**
4.2 Of Skills	**-**

5. Conditions (if necessary)

5.1 For Course Operation	**-**
5.2 For Seminary/Laboratory Operation	**-**

6. Specific Skills Acquired

Professional Skills	**C1. The description of concepts, theories, and models used in the target domain.** **C2. The use of mathematical and computer science models and tools for solving some specific computer graphics problems.** **C3. The writing of source codes and the unitary testing of software components in a known programming language, based on some given design specification.** **C4. The identification of proper methodologies for developing software systems.**
Transversal Skills	**CT1. The application of the rules for a well-organized and efficient work, developing a responsible attitude towards the teaching and scientific fields, in order to achieve the creative capitalization of the student's own potential, by respecting the principles and rules of professional ethics.** **CT2. The use of efficient methods and techniques for learning, acquiring information, research and development of the capabilities to capitalize the knowledge, to adapt to the requirements of a dynamic society and to communicate in Romanian and in an international language.**

7. Course Objectives (from the grid of specific skills acquired)

7.1 General Objectives	Understanding the underlying principles and hardware/software technologies of the computing systems.
7.2 Specific Objectives	At the end of the semester, the students should know: - the basic elements of the computer architecture and organization - the fundamentals of internal data representation in computers - the main functions of an operating system

8. General Description

8.1	**Course**	**Teaching Methods**	**Observations** (hours and bibliographic references)
1	Introduction. Empirical laws. Combinational circuits and Boolean functions	exposition, debate, case studies, exercises	-
2	Minimization. Combinational circuits	exposition, debate, case studies, exercises	-
3	Sequential circuits	exposition, debate, case studies, exercises	-
4	Internal representations: fixed-point representations	exposition, debate, case studies, exercises	-
5	Fixed-point representations. Overflows. Floating-point representations.	exposition, debate, case studies, exercises	-

6	Floating-point representations. Computer architecture and organization	exposition, debate, case studies, exercises	-
7	The memory. Memory hierarchy. Cache memory	exposition, debate, case studies, exercises	-
8	Recapitulation	exposition, debate, case studies, exercises	-
9	The Central Processing Unit. Pipeline	exposition, debate, case studies, exercises	-
10	Improving the CPU performance	exposition, debate, case studies, exercises	-
11	Paralle architectures. Peripheral devices	exposition, debate, case studies, exercises	-
12	The interrupt system. The operating system. Process management	exposition, debate, case studies, exercises	-
13	Memory management	exposition, debate, case studies, exercises	-
14	Virtual memory. Creation and execution of programs	exposition, debate, case studies, exercises	-

Bibliography

J. L. Hennessy, D. A. Patterson, *Computer Architecture - A Quantitative Approach*, Morgan Kaufmann Publishers, 1990.
D. A. Patterson, J. L. Hennessy, *Computer Organization & Design: The Hardware/Software Interface*, Morgan Kaufmann Publishers, 1998.
R. W. Hockney, C. R. Jesshope, *Parallel Computers 2*, IOP Publishing, 1988.
A. Tanenbaum, *Structured Computer Organization*, Prentice Hall, 1999.
A. Tanenbaum, *Modern Operating Systems*, Prentice Hall, 2001.

8.2	**Seminary / Laboratory**	**Teaching methods**	**Observations** (hours and bibliographic references)
1	Conversions between bases of numeration systems. Amdahl's law	exposition, debate, exercises	-
2	Boolean functions. Minimization	exposition, debate, exercises	-
3	Combinational circuits	exposition, debate, exercises	-
4	Latches and flip-flops	exposition, debate, exercises	-
5	Fixed-point representations	exposition, debate, exercises	-
6	Floating point representations	exposition, debate, exercises	-
7	Introduction to the Intel x86 assembly language	exposition, debate, exercises	-
8	Recapitulation	exposition, debate, exercises	-

9	Arithmetic and logical instructions	exposition, debate, exercises	-
10	Jump instructions	exposition, debate, exercises	-
11	Stack-handling instructions. Function calls	exposition, debate, exercises	-
12	Arrays and pointers	exposition, debate, exercises	-
13	Structures	exposition, debate, exercises	
14	Recapitulatory exercises	exposition, debate, exercises	-

Bibliography

Intel® 64 and IA-32 Architectures Software Developer Manuals:
http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html

9. Course content synchronization with the expectations of the community representatives, professional associations and employers from the program domain

This discipline presents the basic elements of a computing architecture. Such knowledge is essential for understanding the way computers work, regardless of the purpose of their use; particularly, writing efficient programs is not possible without being acquainted with the mechanisms and techniques that are taught here.

10. Evaluation

Activity Type	**10.1 Evaluation criteria**	**10.2 Evaluation methods**	**10.3 The weight of each evaluation form (%)**
10.4 Course	the ability to apply the theoretical knowldege to solving practical problems	two written tests, one for each half of the semester	40% 20%
10.5 Seminary/ Laboratory	the ability to develop programs, emphasizing the importance of efficiency	practical test	40%

10.6 Minimal performance standards

- basic knowledge of the concepts related to the hardware structure of the computing systems
- the ability to develop low-complexity programs in the assembly language of the Intel x86 processor family
- at least 5 points earned on each test; less than 5 points (but no less than 2 points) are accepted for at most one of the tests
- the final result is calculated as the weighted average of the three tests; if less than 5 points were earned at one test, truncation is performed instead of rounding

Date Course Teacher Seminary/Laboratory Teacher

Department Date of Approval Director of the Department