



FIȘA DISCIPLINEI COURSE DESCRIPTION

1. Date despre program

Program Information

1.1. Instituția de învățământ superior <i>University</i>	Universitatea “Alexandru Ioan Cuza” din Iași <i>“Alexandru Ioan Cuza” University of Iași</i>
1.2. Facultatea <i>Faculty</i>	Facultatea de Informatică <i>Faculty of Computer Science</i>
1.3. Departamentul <i>Department</i>	Informatică <i>Department of Computer Science</i>
1.4. Domeniul de studii <i>Study Domain</i>	Informatică <i>Computer Science</i>
1.5. Ciclul de studii <i>Study Cycle</i>	Licență <i>Undergraduate studies</i>
1.6. Programul de studii / Calificarea <i>Study Program / Qualification</i>	Informatică / Licențiat în Informatică <i>Computer Science / Computer Science Graduate</i>

2. Date despre disciplină

Course Information

2.1. Denumirea disciplinei <i>Course Name</i>	Arhitectura calculatoarelor și sisteme de operare <i>Computer Architecture and Operating Systems</i>						
2.2. Titularul activităților de curs <i>Course Teacher</i>	Lect. dr. Vlad Rădulescu						
2.3. Titularul activităților de seminar <i>Seminary Teacher</i>	Lect. dr. Vlad Rădulescu						
2.4. An de studiu <i>Year of study</i>	1	2.5. Semestru <i>Semester</i>	1	2.6. Tip de evaluare <i>Evaluation</i>	E	2.7. Regimul disciplinei <i>Course status</i>	OB

* OB – Obligatoriu / OP – Opțional

* OB – Compulsory / OP – Optional

3. Timpul total estimat (ore pe semestru și activități didactice)

Total estimated hours (hours per semester and didactic activities)

3.1. Număr de ore pe săptămână <i>Hours per week</i>	4	din care: 3.2. curs <i>of which: course</i>	2	3.3. seminar/laborator <i>seminary/laboratory</i>	2
3.4. Total ore din planul de învățământ <i>Hours in curriculum</i>	56	din care: 3.5. curs <i>of which: course</i>	28	3.6. seminar/laborator <i>seminary/laboratory</i>	28
Distribuția fondului de timp <i>Time Distribution</i>					Ore <i>Hours</i>
Studiu după manual, suport de curs, bibliografie și altele <i>Manual study, Course support, Bibliography, and others</i>					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren <i>Supplementary Documentation in library, on electronic forums, and in the field</i>					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri <i>Seminaries/laboratories preparation, homeworks, reports, portfolios, and essays</i>					25
Tutoriat <i>Tutoring</i>					5
Examinări <i>Evaluation</i>					4
Alte activități <i>Other activities (consultations per student)</i>					5
3.7. Total ore studiu individual <i>Total hours individual study</i>					55
3.8. Total ore pe semestru <i>Total hours per semester</i>					125



3.9. Număr de credite
Credits

5

4. **Precondiții** (dacă este cazul)
Preconditions (if necessary)

4.1. De curriculum <i>Of Curriculum</i>	-
4.2. De competențe <i>Of Skills</i>	-

5. **Condiții** (dacă este cazul)
Conditions (if necessary)

5.1. De desfășurare a cursului <i>For Course Operation</i>	-
5.2. De desfășurare a seminarului/ laboratorului <i>For Seminary/Laboratory Operation</i>	Prezența este obligatorie la laborator. <i>Attending the seminary/laboratory classes is mandatory.</i>

6. **Competențe specifice acumulate**
Specific Skills Acquired

Competențe profesionale <i>Professional Skills</i>	La finalizarea cu succes a acestei discipline, studenții vor fi capabili: C1. Să descrie conceptele, teoriile și modelele folosite în domeniul proiectării hardware. C2. Să utilizeze modelele și instrumentele informatice și matematice pentru rezolvarea problemelor specifice domeniului proiectării hardware. C3. Să scrie cod sursă adecvat într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date. C4. Să identifice metodologii adecvate de dezvoltare a produselor software. <i>Upon successful completion of this discipline, students will be able to:</i> C1. Describe the concepts, theories, and models employed in the field of hardware design. C2. Use mathematical models and tools for solving problems that are specific to hardware design. C3. Write appropriate source code in a programming language, based on the given design specifications. C4. Identify the appropriate development methodologies for the software products.
Competențe transversale <i>Transversal Skills</i>	La finalizarea cu succes a acestei discipline, studenții vor fi capabili: C1. Să combine cunoștințele dobândite din diferite surse, să realizeze analogii între problemele și tehnicile de rezolvare din diferite domenii. C2. Să abordeze probleme interdisciplinare. C3. Să utilizeze metode eficiente de învățare, informare, cercetare. <i>Upon successful completion of this discipline, students will be able to:</i> C1. Combine the knowledge acquired from various sources and to make analogies between the problems and solving techniques from various fields. C2. Approach interdisciplinary problems. C3. Use efficient techniques for learning, acquiring information, and research.

7. **Obiectivele disciplinei** (din grila competențelor specifice acumulate)
Course Objectives (from the grid of specific skills acquired)

7.1. Obiectivul general <i>General Objective</i>	Înțelegerea principiilor și tehnologiilor hardware și software care stau la baza proiectării sistemelor de calcul. <i>Understanding the underlying principles and hardware/software technologies of the computing systems.</i>
--	---



7.2. Obiectivele specifice <i>Specific Objectives</i>	<p>O1. Cunoașterea elementelor de bază ale arhitecturii și organizării sistemelor de calcul. O2. Cunoașterea principiilor fundamentale ale reprezentării interne a datelor într-un sistem de calcul. O3. Cunoașterea funcțiilor principale ale sistemelor de operare. O4. Cunoașterea elementelor de bază ale limbajului de asamblare al unui procesor.</p> <p><i>O1. Knowing the basic elements of the computer architecture and organization. O2. Knowing the fundamentals of internal data representation in computing systems. O3. Knowing the main functions of an operating system. O4. Knowing the basic elements of the assembly language of a processor.</i></p>
---	---

8. Conținut

General Description

8.1.	Curs <i>Course</i>	Metode de predare <i>Teaching methods</i>	Observații (ore și referințe bibliografice) <i>Observations</i> (hours and bibliographic references)
1	Circuite combinaționale și funcții booleene. Minimizare. <i>Combinational circuits and Boole functions. Minimization.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
2	Circuite combinaționale complexe. Circuite secvențiale. <i>Complex combinational circuits. Sequential circuits.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
3	Circuite secvențiale complexe. Reprezentări interne. <i>Complex sequential circuits. Internal representations.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
4	Reprezentări în virgulă fixă. Depășiri. <i>Fixed-point representations. Overflow.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
5	Reprezentări în virgulă mobilă. Arhitectura calculatorului. <i>Floating-point representations. Computer architecture.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
6	Organizarea calculatorului. Unitatea centrală de procesare. Memoria. <i>Computer organization. The central processing unit. The memory.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
7	Memoria cache. Sistemul de întreruperi. <i>Cache memory. The interrupt system.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
8	Recapitulare. <i>Recapitulation.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
9	Sistemul de operare. Apeluri sistem. Gestiunea memoriei. Introducere în limbajul de asamblare. <i>The operating system. System calls.</i>	expunere, dezbateri, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-



	<i>Memory management. Assembly language - introduction.</i>		
10	Segmentarea memoriei. Limbajul de asamblare - instrucțiuni aritmetice și logice. <i>Memory segmentation. Assembly language - arithmetic and logical instructions.</i>	expunere, dezbatere, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
11	Paginarea memoriei. Limbajul de asamblare - instrucțiuni de salt; structuri de control. <i>Memory pagination. Assembly language - branch instructions; flow control structures.</i>	expunere, dezbatere, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
12	Gestiunea proceselor. Comunicare între procese. Limbajul de asamblare - lucrul cu stiva; apeluri de funcții. <i>Process management. Assembly language - the stack; function calls.</i>	expunere, dezbatere, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
13	Crearea și execuția programelor. Limbajul de asamblare - tablouri și pointeri; structuri. <i>Creation and execution of programs. Assembly language - arrays and pointers; structures.</i>	expunere, dezbatere, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
14	Optimizarea programelor. Limbajul de asamblare - lucrul în virgulă mobilă. <i>Program optimization. Assembly language - floating point instructions.</i>	expunere, dezbatere, studii de caz, exerciții <i>exposition, debate, case studies, exercises</i>	-
Bibliografie J. L. Hennessy, D. A. Patterson, <i>Computer Architecture - A Quantitative Approach</i> , Morgan Kaufmann Publishers, 1990. D. A. Patterson, J. L. Hennessy, <i>Organizarea și proiectarea calculatoarelor. Interfața hardware/software</i> , Ed. All, 2002. R. W. Hockney, C. R. Jesshope, <i>Calculatoare paralele</i> , Ed. Tehnică, 1991. A. Tanenbaum, <i>Organizarea structurată a calculatoarelor</i> , Ed. Agora, 1999. A. Tanenbaum, <i>Modern Operating Systems</i> , Prentice Hall, 2001. L. Vințan, <i>Fundamente ale arhitecturii microprocesoarelor</i> , Ed. Matrix Rom, 2016.			
Bibliography J. L. Hennessy, D. A. Patterson, <i>Computer Architecture - A Quantitative Approach</i> , Morgan Kaufmann Publishers, 1990. D. A. Patterson, J. L. Hennessy, <i>Computer Organization & Design: The Hardware/Software Interface</i> , Morgan Kaufmann Publishers, 1998. R. W. Hockney, C. R. Jesshope, <i>Parallel Computers 2</i> , IOP Publishing, 1988. A. Tanenbaum, <i>Structured Computer Organization</i> , Prentice Hall, 1999. A. Tanenbaum, <i>Modern Operating Systems</i> , Prentice Hall, 2001.			
8.2.	Seminar / Laborator <i>Seminary / Laboratory</i>	Metode de predare <i>Teaching methods</i>	Observații (ore și referințe bibliografice) Observations (hours and bibliographic references)
1	Introducere. Transformări între bazele de numerație. Funcții booleene. <i>Introduction. Conversions between bases of numeration. Boole functions.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-



2	Minimizarea funcțiilor booleene. <i>Boole functions - minimization.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
3	Circuite combinaționale complexe. <i>Complex combinational circuits.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
4	Circuite bistabile. Circuite secvențiale complexe. <i>Latches and flip-flops. Complex sequential circuits.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
5	Reprezentări numerice. Reprezentări în virgulă fixă. <i>Internal number representation. Fixed-point representations.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
6	Reprezentări în virgulă mobilă. <i>Floating point representations.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
7	Legi empirice. Legea lui Amdahl. Memoria cache. <i>Empirical laws. Amdahl's law. Cache memory.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
8	Recapitulare. <i>Recapitulation.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
9	Introducere în limbajul de asamblare Intel x86. <i>Introduction to the Intel x86 assembly language.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
10	Instrucțiuni aritmetice și logice. <i>Arithmetic and logical instructions.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
11	Salturi. Structuri de control. <i>Branch instructions. Flow control structures.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
12	Lucrul cu stiva. Apeluri de funcții. <i>Stack-based instructions. Function calls</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
13	Tablouri și pointeri. Structuri. <i>Arrays and pointers. Structures.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-
14	Instrucțiuni în virgulă mobilă. <i>Floating point instructions.</i>	expunere, dezbatere, exerciții <i>exposition, debate, exercises</i>	-

Bibliografie

Intel® 64 and IA-32 Architectures Software Developer Manuals:

<http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>

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. Coroborarea conținutului disciplinei cu așteptările reprezentanților comunității, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

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

Disciplina de față prezintă studenților elementele de bază ale unei arhitecturi de calcul. Aceste cunoștințe sunt esențiale pentru înțelegerea modului de lucru al calculatoarelor, indiferent de scopul pentru care sunt folosite; în particular, scrierea de programe performante nu este posibilă fără a cunoaște mecanismele și tehnicile predate aici.

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; in particular, writing efficient



programs is not possible without being acquainted with the mechanisms and techniques that are taught here.

10. Evaluare Evaluation

Tip activitate Activity Type	10.1.Criterii de evaluare Evaluation criteria	10.2.Metode de evaluare Evaluation methods	10.3.Pondere în nota finală (%) Weight per evaluation form (%)
10.4.Curs Course	<ul style="list-style-type: none">- Înțelegerea conceptelor teoretice de bază privind proiectarea hardware și reprezentările numerice.- Capacitatea de a aplica elementele teoretice la rezolvarea de probleme de natură practică.- <i>Understanding the basic theoretical concepts of hardware design and representations.</i>- <i>The ability to apply the theoretical knowledge to solving practical problems.</i>	1 test scris, la finalul primei jumătăți de semestru <i>1 written test, after the first half of the semester</i>	50%
10.5.Seminar/ Laborator Seminary/ Laboratory	Capacitatea de a dezvolta programe, cu accent pus pe eficiența programării. <i>The ability to develop programs, emphasizing the importance of efficiency.</i>	1 test practic <i>1 practical test</i>	50%
10.6.Standard minim de performanță <i>Minimal performance standards</i>			
<ul style="list-style-type: none">➤ Cunoașterea la nivel minimal a conceptelor legate de hardware-ul sistemelor de calcul.➤ Capacitatea de a dezvolta programe de complexitate redusă în limbajul de asamblare al familiei de procesoare Intel x86.➤ <i>Basic knowledge of the concepts related to the hardware structure of the computing systems.</i>➤ <i>The ability to develop low-complexity programs in the assembly language of the Intel x86 processor family.</i>			

Data completării
Date

Titular de curs și seminar
Course and Seminary/Laboratory
Teacher

30.09.2024

Data avizării în departament
Department Date of Approval

Director de departament
Director of the Department