

**FIȘA DISCIPLINEI****1. Date despre program**

1.1 Instituția de învățământ superior	Universitatea "Alexandru Ioan Cuza" din Iași
1.2 Facultatea	Facultatea de Informatică
1.3 Departamentul	DEPARTAMENTUL DE INFORMATICA
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Inteligență artificială

**2. Date despre disciplină**

2.1 Denumirea disciplinei	Arhitectura calculatoarelor și sistemelor de operare						
2.2 Titularul activităților de curs	Lect. Dr. VLAD TUDOR RADULESCU						
2.3 Titularul activităților de seminar	Asist. GHEORGHE BALAN/ Asist. Dr. VLAD-CONSTANTIN CRACIUN						
2.4 An de studiu	I	2.5 Semestrul	I	2.6 Tip de evaluare *	E	2.7 Regimul disciplinei**	Ob

\*E – Examen / C – Colocviu / V – Verificare

\*\*OB – Obligatoriu / OP – Opțional / F – Facultativ

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

3.1 Număr de ore pe săptămână	4	3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și altele					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminare/laboratoare, teme, referate, portofolii și eseuri					15
Tutoriat					5
Examinări					4
Alte activități					0
3.7 Total ore studiu individual*					69
3.8 Total ore pe semestru					125
3.9 Numărul de credite					5

**4. Precondiții - De curriculum (dacă este cazul)**

-
---

## 5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	-
5.2 De desfășurare a seminarului/ laboratorului	-

## 6. Obiective

Înțelegerea principiilor și tehnologiilor hardware și software care stau la baza dezvoltării sistemelor de calcul.

## 7. Competențe/Rezultate ale învățării

- Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.
- Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
- Studentul/absolventul coordonează echipe tehnice pentru dezvoltarea de aplicații informatice, asumând decizii responsabile legate de optimizarea și integrarea acestora.

## 8. Conținut

8.1 Curs	Metode de predare	Observații (ore și referințe bibliografice)
Introducere. Circuite combinaționale și funcții booleene	expunere, dezbateri, studii de caz, exerciții	-
Minimizare. Circuite combinaționale	expunere, dezbateri, studii de caz, exerciții	-
Circuite secvențiale	expunere, dezbateri, studii de caz, exerciții	-
Reprezentări interne. Reprezentări în virgulă fixă	expunere, dezbateri, studii de caz, exerciții	-
Reprezentări în virgulă fixă. Depășiri. Reprezentări în virgulă mobilă	expunere, dezbateri, studii de caz, exerciții	-
Reprezentări în virgulă mobilă. Arhitectura și organizarea calculatorului	expunere, dezbateri, studii de caz, exerciții	-
Memoria. Ierarhia de memorii. Memoria cache	expunere, dezbateri, studii de caz, exerciții	-
Recapitulare	expunere, dezbateri, studii de caz, exerciții	-
Unitatea centrală de procesare. Pipeline	expunere, dezbateri, studii de caz, exerciții	-
Îmbunătățirea performanței procesoarelor	expunere, dezbateri, studii de caz, exerciții	-
Arhitecturi paralele de calcul. Dispozitive periferice	expunere, dezbateri, studii de caz, exerciții	-
Sistemul de întreruperi. Sistemul de operare. Gestiunea proceselor	expunere, dezbateri, studii de caz, exerciții	-

8.1 Curs	Metode de predare	Observații (ore și referințe bibliografice)
Gestiunea memoriei	expunere, dezbateri, studii de caz, exerciții	-
Memoria virtuală. Crearea și execuția programelor	expunere, dezbateri, studii de caz, exerciții	-

### Bibliografie

J. L. Hennessy, D. A. Patterson, Computer Architecture - A Quantitative Approach, Morgan Kaufmann Publishers, 1990.  
D. A. Patterson, J. L. Hennessy, Organizarea și proiectarea calculatoarelor. Interfața hardware/software, Ed. All, 2002.  
R. W. Hockney, C. R. Jesshope, Calculatoare paralele, Ed. Tehnică, 1991.  
A. Tanenbaum, Organizarea structurată a calculatoarelor, Ed. Agora, 1999.  
A. Tanenbaum, Modern Operating Systems, Prentice Hall, 2001.  
L. Vințan, Fundamente ale arhitecturii microprocesoarelor, ed. Matrix Rom, 2016.

8.2 Seminar / Laborator	Metode de predare	Observații (ore și referințe bibliografice)
Introducere. Transformări între bazele de numerație	expunere, dezbateri, exerciții	-
Minimizarea funcțiilor booleene	expunere, dezbateri, exerciții	-
Circuite combinaționale	expunere, dezbateri, exerciții	-
Circuite bistabile	expunere, dezbateri, exerciții	-
Reprezentări în virgulă fixă	expunere, dezbateri, exerciții	-
Reprezentări în virgulă mobilă	expunere, dezbateri, exerciții	-
Introducere în limbajul de asamblare Intel x86	expunere, dezbateri, exerciții	-
Recapitulare	expunere, dezbateri, exerciții	-
Instrucțiuni aritmetice și logice	expunere, dezbateri, exerciții	-
Instrucțiuni de salt	expunere, dezbateri, exerciții	-
Lucrul cu stiva. Apeluri de funcții	expunere, dezbateri, exerciții	-
Tablouri și pointeri. Structuri	expunere, dezbateri, exerciții	-
Instrucțiuni în virgulă mobilă	expunere, dezbateri, exerciții	-
Exerciții recapitulative	expunere, dezbateri, exerciții	-

### Bibliografie

J. L. Hennessy, D. A. Patterson, Computer Architecture - A Quantitative Approach, Morgan Kaufmann Publishers, 1990.  
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

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.

## 10. Evaluare

10.1 Evaluare continuă	Pondere (min. 30%)	80
------------------------	--------------------	----

Curs	Forma de evaluare			
	Pondere		0	
	Nepromovarea Evaluării continue determină nepromovarea Evaluării finale			
	Metode de evaluare	Detalii	Pondere	cu reexaminare
Seminar / Laborator	Forma de evaluare		Verificare practică	
	Pondere		100	
	Nepromovarea Evaluării continue determină nepromovarea Evaluării finale		Nu	
	Metode de evaluare	Detalii	Pondere	cu reexaminare
		Test	50	Da
Test practic		50	Nu	

<b>10.2 Evaluare finală</b>	Pondere (max. 70%)	20
	Forma de evaluare	Verificare scrisă finală

<b>10.3 Mențiuni (situații speciale în evaluare)</b>	
-	

<b>10.4 Standard minim de performanță</b>
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.

**Data  
completării,**

**Titular de curs,  
Lect. Dr. VLAD TUDOR  
RADULESCU**

**Titular de seminar,  
Asist. GHEORGHE BALAN/ Asist. Dr. VLAD-CONSTANTIN  
CRACIUN**

**Data avizării în departament,**

**Director de departament,  
Conf. Dr. ANDREI ARUSOAIIE**