

## FI A DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematica și Informatică</b>
1.3 Departamentul	<b>Departamentul de Informatică</b>
1.4 Domeniul de studii	<b>Matematică</b>
1.5 Ciclul de studii	<b>Licență</b>
1.6 Programul de studiu / Calificarea	<b>Matematică Informatică - română</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Arhitectura sistemelor de calcul</b>						
2.2 Titularul activităților de curs	<b>Lect. Dr. Anca Andreica</b>						
2.3 Titularul activităților de seminar	<b>Lect. Dr. Anca Andreica</b>						
2.4 Anul de studiu	2	2.5 Semestrul	3	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	<b>Obligatorie</b>

### 3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator	1 sem + 1 lab
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					20
Tutoriat					5
Examinări					14
Alte activități: .....					
3.7 Total ore studiu individual		69			
3.8 Total ore pe semestru		125			
3.9 Numărul de credite		5			

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	•
4.2 de competențe	•

### 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	•
5.2 De desfășurare a seminarului/laboratorului	• Laborator cu calculatoare

## 6. Competențele specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> <li>• Cunoașterea, înțelegerea și folosirea conceptelor de bază ale informaticii</li> </ul>
Competențe transversale	<ul style="list-style-type: none"> <li>• Cunoașterea modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, utilizarea sistemelor de reprezentare a informației în calculator.</li> <li>• Inițiere în programarea în limbaj de asamblare, ceea ce asigură înțelegerea arhitecturii și funcționării unui microprocesor.</li> <li>• Inițiere în arhitecturile sistemelor de întreruperi, cu particularizarea la mașinile 80x86</li> </ul>

## 7. Obiectivele disciplinei (reie îndin grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Cunoașterea modelelor arhitecturale ale calculatoarelor, funcționarea procesorului, utilizarea sistemelor de reprezentare a informației în calculator.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Inițiere în programarea în limbaj de asamblare, ceea ce asigură înțelegerea arhitecturii și funcționării unui microprocesor.</li> <li>• Inițiere în arhitecturile sistemelor de întreruperi, cu particularizarea la mașinile 80x86.</li> <li>• Conștientizarea influenței pe care principiile funcționale de bază ale arhitecturii von Neumann le au asupra modului de implementare a limbajelor de programare de nivel înalt</li> <li>• Conștientizarea impactului arhitectural asupra tehnicilor de proiectare și implementare a limbajelor de programare de nivel înalt.</li> </ul>

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
1-2. Reprezentarea datelor: date elementare, reprezentări binare și ordini de plasare, organizarea și memorarea datelor, codificarea caracterelor, reprezentarea cu semn și fără semn, cod complementar, conversii, conceptul de depășire	Expunerea, conversația, dezbaterile, problematizarea, descoperirea	
3-4. Arhitectura sistemelor de calcul: organizarea unui	Expunerea, conversația,	

SC, unitatea central , ceasul sistem, calculator pe n bi i, memoria, dispozitivele periferice, performan ele unui SC, arhitectura microprocesorului 80x86 – structur , regi tri, calculul de adres , moduri de adresare, adrese far i near	dezbaterea, problematizarea, descoperirea	
5. Elementele limbajului de asamblare: formatul unei linii surs , expresii, tipuri de accesare a operanzilor, operatori, directive pentru definirea segmentelor, pt.definirea datelor, LABEL, EQU, PROC, INCLUDE, blocuri repetitive i macrouri	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	
6-7. Instruc iuni ale limbajului de asamblare: instruc iuni de transfer, conversii, opera ii aritmetice cu semn i f r semn, opera ii de deplasare i rotire de bi i, opera ii logice pe bi i, instruc iuni de salt condi ionat i necondi ionat, instruc iuni de ciclare, instruc iuni pe iruri	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	
8-9. Întreruperi: clasificare, instruc iuni specifice lucrului cu întreruperi, formatele COM i EXE; Redirectarea întreruperilor: programe TSR, instalarea i deinstalarea programelor TSR, depanarea programelor TSR, redirectarea întreruperilor în cadrul SO Windows	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	
10-11. Implementarea apelului de subprograme i programare multimodul: cod de apel, cod de intrare, cod de ie ire, directivele PUBLIC, EXTRN, GLOBAL, legarea de module TASM cu module scrise în limbaje de nivel înalt	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	
12. Programare low-level în limbaje de nivel înalt: inserare de cod ma in , asamblare inline, proceduri i func ii assembler, accesarea regi trilor i apelarea de întreruperi, proceduri i func ii interrupt	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	
13. Extensii x86: modul de lucru protejat, extensii arhitecturale i noi instruc iuni introduse la nivelul evolu iei familiei de procesoare 80x86	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	
14. Programarea în limbaj de asamblare sub Windows: apeluri sistem în modul de lucru protejat, limit ri ale sistemului de întreruperi, asamblarele MASM i NASM, asamblorul inline Visual C++	Expunerea, conversa ia, dezbaterea, problematizarea, descoperirea	

#### Bibliografie:

1. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
2. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005.
3. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003.  
(<http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html>)
4. Boian F. M. Sisteme de operare interactive. Ed. Libris, Cluj, 1994

<p>5. Boian F. M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996</p> <p>6. Boian F. M., Vancea A., Iurian S., Iurian M. Programare avansata de sistem si aplicatii IBM-PC, lito. Universitatea "Babes-Bolyai", 1996</p> <p>7. Boian F.M. Vancea A. Arhitectura calculatoarelor, suport de curs. Facultatea de Matematica si Informatica, Centrul de Formare Continua si Invatamânt la Distanta., Ed. Centrului de Formare Continua si Invatamânt la Distanta, Cluj, 2002,</p> <p>8. Knuth D.E. Tratat de programarea calculatoarelor; vol 3: Algoritmi seminumerici. Ed. Tehnica, Bucuresti, 1985</p>		
8.2 Seminar / laborator	Metode de predare	Observa ii
1. Reprezentarea datelor, conversii, conceptul de dep ire	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	Exista cate un seminar la fiecare doua saptamani, si cate un laborator la fiecare doua saptamani; materia predata la seminar este coroborata cu activitatea de la laborator
2. Instruc iuni ale limbajului de asamblare: instruc iuni de transfer, conversii, opera ii aritmetice cu semn i f r semn	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	
3. Opera ii de deplasare i rotire de bi i, opera ii logice pe bi i	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	
4. Instruc iuni de salt condi ionat i necondi ionat, instruc iuni de ciclare, instruc iuni pe iruri	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	
5. Intreruperi, instruc iuni specifice lucrului cu îtreruperi	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	
6. Implementarea apelului de subprograme i programare multimodul	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	
7. Programare low-level în limbaje de nivel înalt	Dialogul, dezbaterea, studiul de caz, exemple, demonstratii	
<p><b>Bibliografie:</b></p> <p>1. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.</p> <p>2. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005.</p> <p>3. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003. (<a href="http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html">http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/DOS/index.html</a>)</p>		

**9. Coroborarea con inuturilor disciplinei cu a tept rile reprezentan ilor comunit ii epistemice, asocia iilor profesionale i angajatori reprezentativi din domeniul aferent programului**

- Acest curs exista in programul de studiu al tuturor universitatilor importante din Romania si strainatate
- Acest curs asigura cunostintele de baza pe care orice programator trebuie sa la aiba

## 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota final
10.4 Curs	- cunoasterea principiilor de baza ale domeniului	Examen scris	60%
	- aplicarea acestor concepte in rezolvarea de probleme		
10.5 Seminar/laborator	- implementarea in limbaj de asamblare	Teme laborator	20%
		Examen practic	20%
10.6 Standard minim de performan			
<ul style="list-style-type: none"><li>• Pentru promovare este necesara obtinerea notei minim 5 la examenul scris, examenul practic si temele de laborator</li></ul>			

Data completării

24.04.2013

Titular de curs

Lect. Dr. Anca Andreica

Titular de seminar

Lect. Dr. Anca Andreica

Data avizării în departament

.....

Director de departament

.....