

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Matematică
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Matematică Informatică (în limba română)

2. Date despre disciplină

2.1 Denumirea disciplinei	Arhitectura sistemelor de calcul						
2.2 Titularul activităților de curs	Prof. Dr. Anca Andreica						
2.3 Titularul activităților de seminar	Prof. Dr. Anca Andreica						
2.4 Anul de studiu	2	2.5 Semestrul	3	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligativu

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					5
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					10
Tutoriat					5
Examinări					4
Alte activități:					
3.7 Total ore studiu individual	44				
3.8 Total ore pe semestru	100				
3.9 Numărul de credite	4				

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	<p>C6.1 Identificarea conceptelor și modelelor de bază pentru sisteme de calcul și rețele de calculatoare.</p> <p>C6.2 Identificarea și explicarea arhitecturilor de bază pentru organizarea și gestiunea sistemelor și a rețelelor.</p>
Competențe transversale	<p>CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

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

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1-2. Reprezentarea datelor	Expunerea, conversația, dezbateră, problematizarea, descoperirea	
3-4. Arhitectura microprocesorului 80x86	Expunerea, conversația, dezbateră, problematizarea, descoperirea	

5-7. Elementele limbajului de asamblare	Expunerea, conversația, dezbateră, problematizarea, descoperirea	
8-10. Instrucțiuni ale limbajului de asamblare	Expunerea, conversația, dezbateră, problematizarea, descoperirea	
11. Întreruperi, redirectarea întreruperilor	Expunerea, conversația, dezbateră, problematizarea, descoperirea	
12-13. Implementarea apelului de subprograme și programare multimodul	Expunerea, conversația, dezbateră, problematizarea, descoperirea	
14. Programare low-level în limbaje de nivel înalt	Expunerea, conversația, dezbateră, 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
5. Boian F. M. De la aritmetica la calculatoare. Ed. Presa Universitara Clujeana, Cluj, 1996
6. Boian F. M., Vancea A., Iurian S., Iurian M. Programare avansata de sistem si aplicatii IBM-PC, lito. Universitatea "Babes-Bolyai", 1996
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,
8. Knuth D.E. Tratat de programarea calculatoarelor; vol 3: Algoritmi seminumerici. Ed. Tehnica, Bucuresti, 1985

8.2 Seminar / laborator	Metode de predare	Observații
1. Reprezentarea datelor, conversii, conceptul de depășire	Dialogul, dezbateră, studiul de caz, exemple, demonstratii	Exista cate un seminar la fiecare doua saptamani, si cate un laborator la fiecare doua saptamani; materia predată la seminar este coroborată cu activitatea de la laborator
2-3. Instrucțiuni ale limbajului de asamblare: instrucțiuni de transfer, conversii, operații aritmetice cu semn și fără semn	Dialogul, dezbateră, studiul de caz, exemple, demonstratii	
4. Operații de deplasare și rotire de biți, operații logice pe biți	Dialogul, dezbateră, studiul de caz, exemple, demonstratii	
5-6. Instrucțiuni de salt condiționat și necondiționat, instrucțiuni de ciclare, instrucțiuni pe șiruri	Dialogul, dezbateră, studiul de caz, exemple, demonstratii	

7. Întreruperi, instrucțiuni specifice lucrului cu întreruperi	Dialogul, dezbateră, studiul de caz, exemple, demonstrații	
Bibliografie:		
<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 și aplicații., Editura Risoprint, Cluj-Napoca, 2005.</p> <p>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)</p>		

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

<ul style="list-style-type: none"> • Acest curs există în programul de studiu al tuturor universităților importante din România și străinătate • Acest curs asigură cunoștințele de bază pe care orice programator trebuie să le aibă

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- cunoașterea principiilor de bază ale domeniului	Examen scris	50%
	- aplicarea acestor concepte în rezolvarea de probleme		
10.5 Seminar/laborator	- implementarea în limbaj de asamblare	Teme laborator	20%
		Activitate seminar	10%
		Examen practic	20%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Pentru promovare este necesară obținerea notei minim 5 la examenul scris, examenul practic și temele de laborator, și prezenta la minim 6 laboratoare și minim 5 seminarii. 			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

.....

Prof. Dr. Anca Andreica

Prof. Dr. Anca Andreica

.....

.....

Data avizării în departament

Semnătura directorului de departament

.....

.....