

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare și Tehnologia Informației/ Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	9.

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare în limbaj de asamblare				
2.2 Titularii de curs	Prof. Dr. ing. Sebestyen-Pal Gheorghe gheorghe.sebestyen@cs.utcluj.ro Conf.dr.ing. Hangan Anca – Anca.Hangan@cs.utcluj.ro				
2.3 Titularul/ Titularii activităților de seminar/laborator/proiect	Conf.dr.ing. Hangan Anca – Anca.Hangan@cs.utcluj.ro , S.l.dr.ing. Neagu Madalin – Madalin.Neagu@aut.utcluj.ro				
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	Curs	2	Seminar		Laborator	2	Proiect	
3.2 Număr de ore pe semestru	56	din care:	Curs	28	Seminar		Laborator	28	Proiect	
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										10
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										17
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										10
(d) Tutoriat										4
(e) Examinări										3
(f) Alte activități:										0
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))							44			
3.5 Total ore pe semestru (3.2+3.4)							100			
3.6 Numărul de credite							4			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Un limbaj de programare (C, Java sau Pascal)
4.2 de competențe	Scrierea de programe într-un limbaj de programare

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator
5.2. de desfășurare a laboratorului	Calculatoare, asamblor, debugger

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C1 - Operarea cu fundamente matematice, ingineresti și ale informaticii</p> <p>C1.1 - Recunoașterea și descrierea conceptelor proprii calculabilității, complexității, paradigmatelor de programare și modelării sistemelor de calcul și comunicații</p> <p>C1.2 - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea structurii și funcționării sistemelor hardware, software și de comunicații</p> <p>C1.3 - Construirea unor modele pentru diferite componente ale sistemelor de calcul</p> <p>C1.4 - Evaluarea formală a caracteristicilor funcționale și nefuncționale ale</p>
-----------------------------	--

	sistemelor de calcul C1.5 - Fundamentarea teoretică a caracteristicilor sistemelor proiectate
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Studierea și însușirea tehnicilor de programare în limbaj de asamblare
7.2 Obiectivele specifice	<ul style="list-style-type: none"> - studierea conceptelor de bază ale unui limbaj de asamblare - studierea elementelor definitorii ale arhitecturii ISA x86 - însușirea limbajului de asamblare al familiei de procesoare Intel x86 - învățarea tehnicilor de programare specifice unui limbaj de nivel scăzut - scrierea de programe în limbaj de asamblare

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere, Reprezentarea informațiilor – tipuri de date, reprezentarea datelor în memorie,		- Mijloace multimedia – Prezentări Power Point	
Stocarea și accesul la date, Arhitectura ISA x86 – tipuri de unități de stocare, elementele de bază ale arhitecturii ISAx86 (registre, indicatoare, moduri de adresare, calculul adreselor de memorie)			
Formatul instrucțiunilor x86, Execuția instrucțiunilor – sintaxa și formatul instrucțiunilor ISAx86, fazele și modurile de execuție ale unei instrucțiuni			
Directivele limbajului de asamblare MASM x86 – directive pentru declararea variabilelor, a constantelor, a segmentelor și a procedurilor, prototip de program scris în limbaj de asamblare			
Setul de instrucțiuni ISA x86 – instrucțiuni de transfer, instrucțiuni aritmetice și logice			
Setul de instrucțiuni ISA x86 (continuare) – instrucțiuni de salt, apeluri de rutine, operații de rotație și deplasare			
Setul de instrucțiuni ISA x86 (continuare) – operații pe șiruri, instrucțiuni de întrerupere, instrucțiuni pe indicatoare de condiție, instrucțiuni 386			
Instrucțiunile coprocesorului matematic – arhitectura coprocesorului matematic, instrucțiuni de transfer, instrucțiuni de conversie			
Instrucțiunile coprocesorului matematic (continuare) instrucțiuni aritmetice în virgula flotantă			
Tehnologia MMX – arhitectura modului MMX, setul de instrucțiuni MMX- Mijloace multimedia – Prezentări Power Point			
Modul de lucru protejat – elementele modului protejat la ISAx86, mecanisme de protecție, calculul adreselor de memorie în modul real și protejat			
Accesul la resursele unui calculator prin funcții sistem – mecanisme de acces la resursele unui calculator, modul de apel al funcțiilor sistem, tipuri de funcții sistem			
Tehnici avansate de programare în limbaj de asamblare – utilizarea procedurilor, implementarea unor operații specifice pe structuri de date complexe			
Tehnici de optimizare a programelor – moduri de evaluare a optimalității, tehnici de creștere a vitezei de procesare și metode de reducere a spațiului de memorie utilizat			
Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>) 1. Curs în format PPT la adresa: bavaria.utcluj.ro/~sebestyen/cursuri_lab.htm 2. D. Gorgan, G. Sebestyen, Proiectarea calculatoarelor”, Editura albastra, 2005, 3. R. Hyde R. Hyde, “AoA - The Art of Assembly language”, la adresa: webster.cs.ucr.edu/AoA/DOS/pdf/ 4. S. Nedevschi, “Microprocesoare”, Editura UTCN, 1994			

8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Reprezentarea informațiilor		Prezentare la tabla, scrierea și depanarea programelor pe calculator	
Arhitectura ISA x86, moduri de adresare, utilitare de depanare			
Moduri de adresare și calculul adreselor de memorie			
Utilizarea directivelor			
Setul de instrucțiuni ISA x86: operații de transfer, aritmetice și logice			
Setul de instrucțiuni ISA x86: operații pe șiruri, apel de rutine			
Setul de instrucțiuni ISA x86: instrucțiuni de salt, alte instrucțiuni			
Operații în virgulă flotantă			
Programarea unor aplicații complexe			
Operații pe date de tip multimedia			
Optimizarea programelor scrise în limbaj de asamblare			
Tehnici de acces la resursele unui calculator			
Tehnici avansate de programare în limbaj de asamblare			
Colocviu			
Bibliografie (bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător)			
1. D. Gorgan, G. Sebestyen, Proiectarea calculatoarelor”, Editura albastra, 2005,			
2. R. Hyde R. Hyde, “AoA - The Art of Assembly language”, la adresa: webster.cs.ucr.edu/AoA/DOS/pdf/			
3. S. Nedeveschi, “Microprocesoare”, Editura UTCN, 1994			

* Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Conținutul cursului este în acord cu cerințele formulate de firme de profil. La alcatuirea tematicii disciplinei s-au consultat specialiști care utilizează limbajul de asamblare în activitatea curentă (firma BitDefender). Lucrările de laborator au fost actualizate în conformitate cu cerințele actuale și în acord cu evoluția procesoarelor de pe piață.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	- grad de cunoaștere a conceptelor de baza - abilitati de programare în limbaj de asamblare	Examen scris, inclusiv prin tehnici on-line (Moodle, MS Teams și Zoom)	70%
Seminar			
Laborator	- abilitati de scriere a unor functii specifice în limbaj de asamblare - calitatea unui microproiect realizat în cadrul laboratorului	Examen scris și evaluare activitatilor desfășurate pe parcursul semestrului, folosind inclusiv instrumente on-line (Moodle, MS Teams și Zoom)	30%
Proiect			
Standard minim de performanță: scrierea de secvențe de program (rutine) de complexitate medie (20-30 instrucțiuni/rutina). Calcul nota disciplina: 70% laborator + 30% examen final Condiții de participare la examenul final: Laborator ≥ 5 Condiții de promovare: Examen final ≥ 5			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
Curs		Prof. Gheorghe Sebestyen-Pal	
		Conf.dr.ing. Anca Hangan	
Aplicații		Conf.dr.ing. Anca Hangan	
		S.I.dr.ing. Madalin Neagu	

Data avizării în Consiliul Departamentului

Director Departament
Prof.dr.ing. Rodica Potolea

Data aprobării în Consiliul Facultății de Automatică și Calculatoare

Decan
Prof.dr.ing. Liviu Miclea