

Anexa 1. Fișa disciplinei R40 – F01

FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul	Departamentul de Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licență - Dual
Programul de studii/calificarea	Calculatoare / Inginer

2. Date despre disciplină

Denumirea disciplinei	PROGRAMAREA CALCULATOARELOR SI LIMBAJE DE PROGRAMARE III				
Anul de studiu	II	Semestrul	IV	Tipul de evaluare	Examen
Regimul disciplinei	Categoría formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categoría de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				DOB

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	Total genera l	4	Curs	2	Seminar	Laborator IIS	2	Proiect IIS		Practică IIS	
						Laborator IM		Proiect IM		Practică IM	
I b) Totalul de ore pe semestru din planul de învățământ		56	Curs	28	Seminar	Laborator	28	Proiect		Practică	

(IIS – instituție de învățământ superior; IM – învățare prin muncă)

Distribuția fondului de timp pe semestru	Ore IIS	Ore IM
II.a) Studiu individual	41	47
II.b) Tutoriat (pentru ID)		
III. Examinări	3	
IV. Alte activități (precizați):		

Total ore studiu individual II + III	Ore IIS	44	Ore IM	
Total ore pe semestru (Ib+II+III+IV)	Ore IIS	100	Ore IM	
Numărul de credite	Credite IIS	4	Credite IM	

4. Competențe specifice acumulate

Competențe profesionale	CP3-aliniaza software-ul la arhitecturile de sistem; CP5-creeaza softuri;
Competențe transversale	

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare, baze de date, inteligență artificială și inginerie software și modul lor de aplicare în probleme concrete	Studentul/absolventul specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), aplicând elementele specifice ingineriei software.	Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei

6. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	- Dezvoltarea de aplicații utilizând limbajul de asamblare. - Manifestarea interesului față de programarea în limbaj de asamblare.
Obiective specifice	- Prezentarea limbajului de asamblare aferente familiei de procesoare I8086. - În cadrul orelor de laborator studenții sunt familiarizați cu diverse metode de programare, utilizând resursele și particularitățile microprocesoarelor studiate. - Se urmărește însușirea principalelor modalități de realizare practică a unor programe editate direct în Assembler sub diferite aplicații cum ar fi sub comanda DOS Debug sau Turbo-Assembler (TASM) și Macro-Assembler (MASM).

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observații
1. Introducere. 1.1 Arhitectura procesorului I8086. 1.2 Unitatea de execuție. 1.3 Unitatea de interfata cu magistrala. 1.4 Registre de uz general. 1.5 Moduri de adresare.	2h	expunerea, prelegerea-dezbateri, demonstrația	
2. Setul de instrucțiuni 8086. 2.1 Instrucțiuni de transfer. 2.2 Instrucțiuni aritmetice și logice. 2.3 Instrucțiuni pentru operații cu siruri de caractere/cuvinte. 2.4 Instrucțiuni de apel de procedură și de salt. 2.5 Instrucțiuni pentru controlul procesorului. 2.6 Dezvoltarea programelor în limbaj de asamblare.	4h	expunerea, prelegerea-dezbateri, demonstrația	
3. Structura programelor. Definirea și initializarea datelor. Operatori. 3.1 Segmentare. Directive pentru definirea segmentelor. 3.2 Directive pentru legarea modulelor. 3.3 Definirea și initializarea datelor. 3.4 Definirea etichetelor. 3.5 Definirea structurilor. 3.6 Operatori în limbajul de asamblare. 3.7 Directive de asamblare conditionată.	2h	expunerea, prelegerea-dezbateri, demonstrația	
4. Macroinstrucțiuni. 4.1 Scopul macroinstrucțiunilor. Definire și expandare. 4.2 Macroinstrucțiuni cu parametri. 4.3 Macroinstrucțiuni repetitive. 4.4 Invocarea recursivă. 4.5 Tehnici avansate de utilizare a macroinstrucțiunilor.	2h	expunerea, prelegerea-dezbateri, demonstrația	
5. Tehnici de programare în limbaj de asamblare. 5.1 Decizia simplă și decizia compusă. 5.2 Cicluri cu test la partea superioară și inferioară. 5.3 Selecția. Tabele de salt sau de apel de proceduri.	6h	expunerea, prelegerea-dezbateri, demonstrația	

5.4 Transferul parametrilor catre proceduri. 5.5 Intoarcerea datelor da catre proceduri. 5.6 Proceduri cu zone de date proprii. 5.7 Proceduri recursive. 5.8 Proceduri cu număr variabil de parametri. 5.9 Tehnici avansate de programare.			
6. Programarea in limbaj de asamblare ARM. 6.1 Manipularea datelor. 6.2 Incarcarea constantelor in registri. 6.3 Incarcarea unei date de memorie intr-un registru. 6.4 Memorarea datelor din registri in memorie. 6.5 Conversia unei instructiuni simple C in limbaj de asamblare ARM. 6.6 Calcularea adreselor de memorie. 6.7 Exemple pentru adresarea memoriei. 6.8 Instructiuni pentru lucrul cu stiva. 6.9 Instructiuni de procesare a datelor.	6h	expunerea, prelegerea-dezbatare, demonstrația	
7. Structurile de control in asamblarea ARM. 7.1 Secventierea instructiunilor. 7.2 Implementarea deciziilor. 7.3 Implementarea buclelor. 7.4 Implementarea functiilor.	6h	expunerea, prelegerea-dezbatare, demonstrația	
Bibliografie minimală recomandată			
1. Practica dezvoltarii software in limbaje de asamblare – Ion Ivan, Paul Pocatilu, Doru Cazan Editura Economica, 2002 2. Totul despre Z80 – M. PATRUBANY – Editura Tehnica, 1989 3. Bazele Microprocesoarelor – Liviu Kreindler – Matrix Rom Bucuresti, 1997 4. Familia de Microcontrolere MCS51 - Vasile Gaitan, 1997 5. Athanasiu I, Panoiu Al - Microprocesoarele 8086,286,386 Ed. TEORA 1992 6. Procesoare Intel, Programare în Limbaj de asamblare, ediția II a, - Vasile Lungu, 2004 7. Programare in limbaj de asamblare, - Gheorghe Musca. Editura Teora. 1998 8. Fundamentals of Embedded Software with the ARM Cortex-M3, Second Edition, Daniel W. Lewis, 2013 9. Modern Arm Assembly Language Programming, Covers Armv8-A 32-bit, 64-bit and SIMD, Daniel Kusswurm, 2020			

Aplicații (laborator)	Nr · or e	Metode de predare	Observații
• Laborator 1. Norme de protecția muncii. Noțiuni introductive ale Programării în Limbaj de Asamblare 2. Reprezentarea datelor. Setul de instrucțiuni 8086. Aritmetica pe 16 biți 3. Aritmetica pe 32 biți. Conversii de lungime a datelor. Instrucțiuni condiționale 4. Tablouri. Operații pe biți 5. Stiva. Subrutine. Macroinstrucțiuni 6. Moduri de transmitere a parametrilor 7. Prelucrarea sirurilor de parametri 8. Structuri. Întreruperi. Evaluarea cunoștințelor 9. Setul de instrucțiuni ARM. 10. Aritmetica pe 16 si 32 de biți 11. Instrucțiuni condiționale ARM 12. Instrucțiuni repetitive ARM 13. Traducere cod C in cod ARM 14. Evaluarea cunoștințelor. Recuperări	2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h 2h	lucrări practice, experimentul	
Bibliografie minimală recomandată			
1. Practica dezvoltarii software in limbaje de asamblare – Ion Ivan, Paul Pocatilu, Doru Cazan Editura Economica, 2002			

2. Totul despre Z80 – M. PATRUBANY – Editura Tehnica, 1989
3. Bazele Microprocesoarelor – Liviu Kreindler – Matrix Rom Bucuresti, 1997
4. Familia de Microcontrolere MCS51 - Vasile Gaitan, 1997
5. Athanasiu I, Panoiu Al - Microprocesoarele 8086,286,386 Ed. TEORA 1992
6. Procesoare Intel, Programare în Limbaj de asamblare, ediția II a, - Vasile Lungu, 2004
7. Programare in limbaj de asamblare, - Gheorghe Musca. Editura Teora. 1998
8. Fundamentals of Embedded Software with the ARM Cortex-M3, Second Edition, Daniel W. Lewis, 2013
9. Modern Arm Assembly Language Programming, Covers Armv8-A 32-bit, 64-bit and SIMD, Daniel Kusswurm, 2020

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	participarea activă în timpul cursurilor.	Evaluare continuă	10 %
	Insusirea notiunilor de baza si utilizarea corecta a limbajului de specialitate	Evaluare prin probă finală de tip test practic (oral) și scris (Moodle) din problemele furnizate la curs.	40%
Seminar			
Laborator/ Lucrări practice	Finalizarea lucrările practice propuse	evaluare continuă (prin metode orale și probe practice)	30%
	Rezolvarea problemelor propuse in temele de acasă	evaluare sumativă (prin prezentarea orală publică, utilizând slide-uri, a rezolvării temei de casă).	20%
Proiect			

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
25.09.2025	Conf. dr.ing Cristian Andy TANASE	As. Ing. Dr. Anda TCACIUC

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
25.09.2025	Conf. dr. ing. Ioan Ungurean

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Prof. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
26.09.2025	Prof. dr. ing. Laurentiu- Dan MILICI