



FIȘA DISCIPLINEI
ANUL UNIVERSITAR 2020 - 2021

1. DATE DESPRE PROGRAM

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN CRAIOVA
1.2 Facultatea	AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
1.3 Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ
1.4 Domeniul de studii	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ
1.5 Ciclul de studii ¹	LICENȚĂ
1.6 Programul de studii (denumire/cod) ² /Calificarea	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ (COD L20601022010)
1.7. Forma de învățământ	CU FRECVENȚĂ

2. DATE DESPRE DISCIPLINĂ

2.1 Denumirea disciplinei	Limbaje de asamblare								
2.2 Titularul activităților de curs	conferențiar doctor Marian Marius Adrian								
2.3 Titularul activităților aplicative	șef lucrări inginer doctor Ganea Ion Eugen								
2.4 Anul de studiu	2	2.5 Semestrul	3	2.6 Tipul disciplinei (conținut) ³	DC	2.7 Regimul disciplinei (obligativitate) ⁴	DFac	2.8 Tipul de evaluare	V

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 laborator	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 laborator	28
3.7 Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și notițe					22
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					6
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					4
Tutoriat					6
Examinări					2
Alte activități: consultații, cercuri studențești					4
Total ore activități individuale	44				
3.8 Total ore pe semestru ⁵	100				
3.9 Numărul de credite ⁶	4				

4. PRECONDIȚII (acolo unde este cazul)

4.1 de curriculum	Studentul trebuie să posede cunoștințe dobândite la următoarele discipline: Analiza matematică, Algebră liniară și geometrie, Metode numerice, Programarea calculatoarelor, Programarea calculatoarelor - Tehnici de programare, Arhitectura calculatoarelor
4.2 de competențe	Cunoștințe generale de programarea calculatoarelor, tehnici de programare și specifice din domeniul arhitecturii calculatoarelor

5. CONDIȚII (acolo unde este cazul)

5.1. de desfășurare a cursului	Predarea cursului se face folosind videoproiectorul. Pentru explicații și răspunsuri la întrebări din sală se folosește tabla. Se asigură suport de curs în format electronic și acces la documentații actualizate. Procesul de predare are următoarea structură:
--------------------------------	---



	80% prezentare teoretică, pe baza suportului de curs (slide-uri) 20% activitate interactivă (discuții cu studenții)
5.2. de desfășurare a seminarului/ laboratorului/proiectului	Laboratorul utilizează o rețea de calculatoare. Sunt dezvoltate aplicații folosind limbaje de asamblare folosindu-se noțiunile referitoare la paradigma programării în limbaje de asamblare prezentate la curs.

6. COMPETENȚELE SPECIFICE ACUMULATE ⁷

Competențe profesionale	Prin cunoștințele predate la curs, prin exemplele prezentate și prin aplicațiile practice efectuate în cadrul laboratorului, cursul „Programarea în limbaje de asamblare” contribuie la formarea competențelor profesionale: Operarea cu fundamente matematice, ingineresti și ale matematicii; Proiectarea componentelor hardware, software și de comunicații; Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor; Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații.
Competențe transversale	

7. OBIECTIVELE DISCIPLINEI (reieșind din grila competențelor specifice acumulate)

7.1 Obiectivul general al disciplinei	Contribuie la formarea viitorilor ingineri din domeniul calculatoarelor și tehnologiei informației, specialiști în dezvoltarea programelor software, asigurându-le cunoștințe în domeniul implementării aplicațiilor în limbaje de asamblare. Sunt abordate concepte de bază și concepte avansate specifice paradigmei de programare în limbaje de asamblare.
7.2 Obiectivele specifice	Introducere în principiile și bunele practici specifice programării în limbaje de asamblare. Utilizarea noțiunilor de bază și avansate pentru a demonstra beneficiile acestora în raport cu criteriile specifice industriei software actuale, de performanță, reutilizabilitate și mentenanță. Laboratorul are rolul de a fixa cunoștințele teoretice și de a crea deprinderi de programare în limbaje de asamblare.

8. CONȚINUTURI

8.1 Curs (unități de conținut)	Nr. ore	Metode de predare
1. Noțiuni de bază. Sisteme de numerație. Reprezentarea datelor	2	Predarea cursului se face folosind videoproiectorul. - 80% prezentare teoretică, pe baza suportului de curs (slide-uri). - 20% activitate interactivă (discuții cu studenții)
2. Elemente de arhitectură. Elemente de memorare a datelor. Registrele procesorului. Arhitectura memoriei. Formatul intern al instrucțiunilor. Moduri de adresare	2	
3. Elemente de bază ale limbajului. Alfabetul limbajului, identificatori, constante. Formatul general al unei instrucțiuni. Variabile. Directive de generare a datelor	2	
4. Setul de instrucțiuni. Instrucțiuni de transfer. Instrucțiuni de interschimbare a datelor. Instrucțiuni de transfer cu stiva. Instrucțiuni de transfer indicatori	2	
5. Setul de instrucțiuni. Instrucțiuni aritmetice. Instrucțiuni de comparare și testare. Înmulțirea și împărțire binară. Instrucțiuni pentru aritmetica BCD	2	



ROMÂNIA
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
UNIVERSITATEA DIN CRAIOVA

FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ
Blvd. Decebal nr.107, Craiova, RO-200440, Tel./Fax +(4)-0251-438.198, <http://ace.ucv.ro>



6. Setul de instrucțiuni. Instrucțiuni de salt. Instrucțiuni de ciclare. Instrucțiuni logice. Instrucțiuni de deplasare și rotire. Instrucțiuni de modificare indicatori	2	Materialele necesare vor fi puse la dispoziția studenților în format electronic și în formă tipărită.
7. Reprezentarea șirurilor de caractere. Prefixe de repetare	2	
8. Instrucțiuni pe șiruri de caractere	2	
9. Proceduri. Stiva și operații cu stiva. Proceduri și transmiterea parametrilor	2	
10. Macroinstrucțiuni. Definierea și utilizarea macroinstrucțiunilor. Directiva LABEL. Utilizarea directivelor de asamblare	2	
11. Macroinstrucțiuni și proceduri. Operatori MACRO. Directive repetitive	2	
12. Structuri și înregistrări pe biți. Definierea unei structuri. Declararea unei variabile de tip structură. Definierea unei înregistrări. Declararea unei variabile de tip înregistrare. Operatorii MASK și WIDTH	2	
13. Programare mixtă. Modele de programare mixtă. Modele de memorie	2	
14. Tratarea întreruperilor utilizând limbaje de asamblare	2	

Bibliografie⁸

1. P.S. Dandamuri, 1998, Introduction to Assembly Language Programming From 8086 to pentium Processors, Springer-Verlag, New York
2. M. Goodwin, 1993, Teach your self assembler, M IS press, New York
3. Gh. Musca, 1998, Programare în limbaj de asamblare, Teora
4. C. Burileanu, M. Ionița, Mihaela Ionița, M. Filotti, 1999, Microprocesoare x86 - o abordare software, Ed. A Ibastra, Cluj-Napoca
5. V. Lungu, 2000, Procesoare Intel, Programarea în limbaje de asamblare, Teora
6. D. Somnea, L. Vladut, 1992, Programarea în Assembler, Editura Tehnică

8.2 Activități aplicative (subiecte/teme)	Nr. ore	Metode de predare
1. Utilizarea funcțiilor DOS pentru afisarea pe ecran a mesajelor	2	Pentru prezentări se va folosi videoproiectorul. - 20% prezentarea noțiunilor teoretice, și a exemplurilor. - 80% activitate interactivă (discuții cu studenții) Materialele necesare vor fi puse la dispoziția studenților în format electronic.
2. Evaluarea unor expresii aritmetice folosind operanzi întregi pe octet și pe cuvânt. Datele sunt specificate în sistemul de numerație zecimal, octal, binar și hexazecimal	2	
3. Însușirea modurilor de adresare a operanzilor. Evaluarea unor expresii aritmetice folosind operanzi întregi pe octet și pe cuvânt adresați prin diverse moduri	2	
4. Probleme care tratează operațiile de înmulțire, împărțire cu și fără semn pe octet și pe cuvânt	2	
5. Adunare și înmulțirea în multiplă precizie	2	
6. Conversia unui număr într-un șir de caractere pentru a fi afișat. Operații cu numere BCD. Implementarea unei structuri de selecție	2	
7. Mini-dezasambloare pentru instrucțiuni de salt condiționat	2	
8. Utilizarea instrucțiunilor pe șiruri de caractere	2	
9. Utilizarea macroinstrucțiunilor	2	
10. Proceduri. Modalități de transmitere a parametrilor	2	
11. Directive de asamblare. Proceduri și macroinstrucțiuni	2	
12. Probleme care tratează programarea mixtă C/C++ - limbaje de asamblare	2	
13. Apelul unor module de program scrise în limbaj de asamblare din module scrise în C/C++	2	
14. Apelul unor module de program scrise în C/C++ din module scrise în limbaj de asamblare	2	

Bibliografie⁸

1. Gheorghe Marian, Marius Marian, Eugen Dumitrașcu, Nicolae Enescu, 2006, Limbaje de asamblare ghid practic, Volumele I și II, Editura Universitaria Craiova
2. Turbo Assembler, Users Guide, Borland International
3. K.A.Lemone, Assembly Language and Systems Programming for the IBM PC and Compatibles", Little Brown & Company Canada Limited
4. Eugen Dumitrașcu, Platforme online pentru laborator, <http://cs.ucv.ro/staff/edumitrascu/PLA.php>



9. COROBORAREA CONȚINUTURILOR DISCIPLINEI CU AȘTEPTĂRILE REPREZENTANȚILOR COMUNITĂȚII EPISTEMICE, ASOCIAȚIILOR PROFESIONALE ȘI ANGAJATORI REPREZENTATIVI DIN DOMENIUL AFERENT PROGRAMULUI

Pentru stabilirea conținuturilor, alegerii metodelor de predare titularul disciplinei a participat la întâlniri cu: reprezentanți mediului de afaceri, precum și cu alte cadre didactice din domeniu. Întâlnirile au vizat identificarea nevoilor și așteptărilor angajatorilor din domeniu și coordonarea cu alte programe similare din cadrul altor instituții de învățământ.

10. EVALUARE

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<ul style="list-style-type: none">- Cunoașterea noțiunilor de bază specifice programării în limbaje asamblare- Capacitatea de utilizare adecvată a noțiunilor de bază pentru proiectarea și dezvoltarea de aplicații software- Înțelegerea importanței exemplelor de implementare în cadrul procesului de	Examen scris final Examen scris intermediar	60% 20%
	dobândire de competențe - Cunoașterea și utilizarea noțiunilor avansate specifice programării în limbaje de asamblare		
10.5 Activități aplicative	<ul style="list-style-type: none">- Însușirea problematicei tratate la curs, seminar și laborator- Capacitatea de a utiliza corect metodele, modelele prezentate- Identificarea și rezolvarea problemelor apărute în cadrul procesului de dezvoltare a aplicațiilor software- Evaluarea temelor de laborator	Evaluare continuă a activității de laborator și seminar	20%
10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)			
<ul style="list-style-type: none">- Cunoașterea și utilizarea noțiunilor de bază specifice programării în limbaje de asamblare;- Identificarea și rezolvarea problemelor apărute în cadrul procesului de dezvoltare a aplicațiilor software			



Data completării: 25.09.2020

**Titular curs
conferențiar dr.ing. Marian M.**

**Titulari activități aplicative
s.l.dr.ing. Ganea Eugen**

Data avizării în departament: : 30.09.2020

**Director de departament
Prof. dr. ing. Cosmin Ionete**