

## FIȘA DISCIPLINEI

### 1. Date despre program

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

### 2. Date despre disciplină

2.1 Denumirea disciplinei	Limbaje formale și tehnici de compilare						
2.2 Titularul activităților de curs	lect. dr. Lupsa Dana						
2.3 Titularul activităților de seminar	lect. dr. Lupsa Dana						
2.4 Anul de studiu	3	2.5 Semestrul	5	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie

### 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					9
Examinări					10
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	<ul style="list-style-type: none"> <li>structuri de date și algoritmi</li> </ul>
4.2 de competențe	<ul style="list-style-type: none"> <li>abilități de programare (nivel mediu)</li> </ul>

## 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"><li>• sala dotata cu videoproiector</li></ul>
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"><li>• laborator dotat cu calculatoare; mediu de dezvoltare pt. limbaje de programare de nivel inalt</li></ul>

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<p>C 4.1 Definirea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice</p> <p>C 4.2 Interpretarea de modele matematice și informatice (formale)</p> <p>C 4.3 Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale</p>
<b>Competențe transversale</b>	<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"><li>• cunoasterea, intelegerea si folosirea conceptelor informatice teoretice de baza folosite in proiectarea compilatoarelor</li><li>• imbunatatirea abilitatilor de programare</li></ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"><li>• cunostinte despre back-end-ul unui compilator</li><li>• imbunatatirea abilitatilor de programare: intelegerea modului in care lucreaza compilatorul, depanarea programelor, o mai buna intelegere a modelului de raportare a erorilor de compilare oferite de compilator (depanarea si corectarea erorilor compilatoarelor)</li><li>• intelegerea conceptelor limbajelor formale si dezvoltarea</li></ul>

	abilitatilor de a modela diverse fenomene folosind limbaje formale ; abilitatea de a aplica tehnici specific compilatoarelor pentru diferite probleme din viata reala
--	---

## 8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Compilator. Introducere.	Expunere, descriere, explicatie, exemplificare	
2. Analiza lexicala. Limbaje. Introducere	Expunere, descriere, explicatie, exemplificare	
3. Gramatica. Clasificarea Chomsky. Automate finite. (part 1)	Expunere, descriere, explicatie, exemplificare	
4. Limbaje regulate. Generatoare de analizoare lexicale	Expunere, descriere, explicatie, exemplificare	
5. Proprietati de inchidere ale limbajelor regulate.	Expunere, explicatie, exemplificare, studiu de caz	
6. Gramatici independente de context	Expunere, explicatie, exemplificare, studiu de caz	
7. Generatoare de analizoare sintactice. Automate Push Down	Expunere, explicatie, exemplificare, studiu de caz	
8,9,10. Analiza sintactica	Expunere, descriere, explicatie, exemplificare	
11. Gramatici de atribute.	Expunere, descriere, explicatie, exemplificare	
12, 13. Cod intermediar. Generarea codului intermediar	Expunere, descriere, explicatie, exemplificare, demonstrare	
14. Aplicativitatea mecanismelor formale studiate in proiectarea compilatoarelor	Expunere: descrierea, exemplificarea	
<b>Bibliografie</b> 1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978. 2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973. 3. D. GRIES - Compiler construction for digital computers,, John Wiley, New York, 1971. 4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006 5. SIPSER, M., Introduction to the theory of computation, PWS Pulb. Co., 1997. 6. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987. 7. <a href="http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC">http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC</a>		
8.2 Seminar	Metode de predare	Observații
1. Specificarea unui limbaj de programare. Notatia BNF	Explicatia, dialogul, studiu de caz	Orele de seminar sint grupate cate 2, la fiecare 2 saptamani
2. Automate finite: <ul style="list-style-type: none"> <li>• limbaj generat de un automat finit, automat finit corespunzator unui limbaj</li> <li>• structuri de date pt. automate finite</li> </ul>	Dialogul, dezbaterea, studiul de caz, exemplificarea	
3. Gramatici, limbaje generate de o gramatica, gramatica corespunzatoare unui limbaj	Dialogul, studiul de caz, exemplificarea, demonstratia	
4. Proprietati ale limbajelor. Demonstratii si aplicatii.	Dialogul, dezbaterea, studiul de caz, exemplificarea,	
5. APD. Gramatici independente de context.	Dialogul, dezbaterea, studiul de caz, exemplificarea	

6. Analiza sintactica LL(1)	Dialogul, studiul de caz, demonstratia	
7. Analiza sintactica LR(k)	Dialogul, dezbaterile, studiul de caz, exemplificarea	
Bibliografie		
1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978. 2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973. 3. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006 4. G. MOLDOVAN, V. CIOBAN, M. LUPEA - Limbaje formale si automate. Culegere de probleme, Univ. Babes-Bolyai, Cluj-Napoca, 1996 5. <a href="http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC">http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC</a>		
8.3 Laborator	Metode de predare	Observatii
1. Tema 1: Specificarea unui mini-limbaj de programare si implementarea unui analizor lexical 1.1 specificarea mini-limbajului de programare	Explicatia, dialogul, studiu de caz	
2. Tema 1: Specificarea unui mini-limbaj de programare si implementarea unui analizor lexical 1.2 implementarea functiilor principale ale analizorului	Explicatia, dialogul, studiu de caz	
3. Tema 1: Specificarea unui mini-limbaj de programare si implementarea unui analizor lexical 1.3 programul principal, testare + predare	Discutarea datelor de test, evaluare	
4. Tema 2: Automate finite 2.1 verificarea acceptarii unei secvente <ul style="list-style-type: none"> <li>• numai pentru AF deterministe</li> <li>• verificati ca AF este determinist</li> </ul>	Explicatia, dialogul, studiu de caz	
5. Tema 2: Automate finite: 2.2 Rescrieti/adaptati programul de analiza lexicala (tema 1) astfel incat sa foloseasca automate finite pentru a determina secventele corespunzatoare atomilor lexicali	Discutarea datelor de test, evaluare	
6. Tema 3: Implementarea unui mini-compiler functional 3.1 specificare si implementare <ul style="list-style-type: none"> <li>• utilizare lex/flex + yacc/bison</li> <li>• refolosire rezolvarea problemelor de la temele 3 si 4</li> <li>• generarea unui program asm echivalent semantic si testarea acestuia</li> </ul>	Explicatia, dialogul, studiu de caz, discutarea datelor de test	
7. Tema3: Implementarea unui mini-compiler functional 3.2 compilarea si testarea programului asm generat testare si corectarea posibilelor erori	Discutarea datelor de test, evaluare	
Bibliografie		
1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978. 2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973. 3. D. GRIES - Compiler construction for digital computers., John Wiley, New York, 1971. 4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006 5. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987. 6. <a href="http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC/">http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC/</a>		

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

- Tematica cursului respecta recomandările de continut IEEE si ACM pentru studiile din domeniul informatica
- Cursul exista in programul de studiu al universitatilor importante din Romania si din alte tari
- Tematica cursului este considerata de companiile soft ca fiind importanta cel putin pentru programatorii de nivel mediu

## 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- cunoasterea principiilor de baza ale domeniului - aplicarea conceptelor studiate la curs - rezolvarea de probleme	Examen scris	75%
10.5 Seminar			
10.6 Laborator	- sa fie capabili sa implementeze conceptele si algoritmi cursului - implementarea unui translator (vazut ca o parte a unui mini-compiler)	Verificare continua a activitatii in timpul orelor de laborator - verificarea documentatiei - verificarea programelor	25%
10.7 Standard minim de performanță			
Prezenta: minim 75% la activitatile de seminar si minim 90% la activitatile de laborator			
Notare:			
<ul style="list-style-type: none"><li>• Cel puțin nota 5 la examenul scris;</li><li>• si cel puțin nota 5 pentru nota pt. laborator.</li></ul>			
De aici rezulta si cel puțin nota 5 pentru media calculata.			

Data completării

.....

Semnătura titularului de curs

Lect. Dr. LUPSA Dana

Semnătura titularului de seminar

Lect. Dr. LUPSA Dana

Data avizării în departament

.....

Semnătura directorului de departament

.....