

## 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	Licenta
1.6 Programul de studiu / Calificarea	Matematică informatică

### 2. Date despre disciplină

2.1 Denumirea disciplinei	Limbașe formale și tehnici de compilare						
2.2 Titularul activităților de curs	conf. dr. Guran Adriana Mihaela						
2.3 Titularul activităților de seminar	conf. dr. Guran Adriana Mihaela						
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	1sem + 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>abilitati 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 înalt</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>• cunoașterea, înțelegerea și folosirea conceptelor informatice teoretice de bază folosite în proiectarea compilatoarelor</li><li>• îmbunătățirea abilităților de programare</li></ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"><li>• cunoștințe despre back-end-ul unui compilator</li><li>• îmbunătățirea abilităților de programare: înțelegerea modului în care lucrează compilatorul, depanarea programelor, o mai bună înțelegere a modelului de raportare a erorilor de compilare oferite de compilator (depanarea și corectarea erorilor compilatoarelor)</li><li>• înțelegerea conceptelor limbajelor formale și 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. Structura generala a unui compilator. Fazele compilarii.	Expunere, descriere, explicatie, exemplificare	
2. Analiza lexicala. Limbaje. Introducere	Expunere, descriere, explicatie, exemplificare	
3. Notiuni introductive pentru limbajele formale. Gramatici si Automate finite	Expunere, descriere, explicatie, exemplificare	
4. Limbaje regulate., expresii regulate, echivalenta dintre automate finite, gramatici regulate si expresii regulate. Lema de pompare	Expunere, descriere, explicatie, exemplificare	
5. Gramatici independente de context, arborele de analiza sintactica	Expunere, explicatie, exemplificare, studiu de caz	
6. Analiza sintactica: notiuni generale, clasificare	Expunere, explicatie, exemplificare, studiu de caz	
7. Analizor sintactic descendent cu reveniri	Expunere, explicatie, exemplificare, studiu de caz	
8. Analizor sintactic LL(1)	Expunere, explicatie, exemplificare, studiu de caz	
9. Analizoare sintactice LR(k). Analizorul sintactic LR(0)	Expunere, explicatie, exemplificare, studiu de caz	
10. Analizor sintactic SLR, LR(1) si LALR	Expunere, descriere, explicatie, exemplificare	
11. Generatoare de analizoare lexicale (lex). Analizoare de analizoare sinactice (yacc)	Expunere, descriere, explicatie, exemplificare	
12. Gramatici de atribute. Cod intermediar. Generarea codului intermediar	Expunere, descriere, explicatie, exemplificare, demonstrare	
14. Automate push-down si Masini Turing	Expunere: descrierea, 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. D. GRIES - Compiler construction for digital computers., John Wiley, New York, 1971. 4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Alabastra, 2006 5. SIPSER, M., Introduction to the theory of computation, PWS Pub. Co., 1997. 6. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987. 7. <a href="http://www.cs.ubbcluj.ro/~adriana/LFTC">http://www.cs.ubbcluj.ro/~adriana/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. Gramatici; limbajul generat de o gramatica; gramatica corespunzatoare unui limbaj de programare	Explicatia, dialogul, studiu de caz	
3. 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, dezbaterile, studiul de caz, exemplificarea	
4. Transformari: automat finit – gramatica regulata-expresie	Dialogul, studiul de caz,	

regularea	exemplificarea, demonstratia	
5. Gramatici independente de context. Analizor sintactic descendent cu reveniri	Dialogul, dezbaterea, studiul de caz, exemplificarea,	
6. Analizor LL(1)	Dialogul, dezbaterea, studiul de caz, exemplificarea	
7. Analizoare LR(0)	Dialogul, studiul de caz, demonstratia	
	Dialogul, dezbaterea, studiul de caz, exemplificarea	
<p>Bibliografie</p> <p>1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.</p> <p>2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973.</p> <p>3. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006</p> <p>4. G. MOLDOVAN, V. CIOBAN, M. LUPEA - Limbaje formale si automate. Culegere de probleme, Univ. Babes-Bolyai, Cluj-Napoca, 1996</p> <p>5. <a href="http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC">http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC</a></p>		
8.3 Laborator	Metode de predare	Observații
1. Tema 1: Specificarea uni mini-limbaj de programare si implementarea unui analizor lexical 1.1 specificarea mini-limbajului de programare in BNF	Explicatia, dialogul, studiu de caz	
2. Tema 1: Specificarea uni mini-limbaj de programare si implementarea unui analizor lexical 1.2 Scrierea unui mini-program in mini-limbajul de programare	Explicatia, dialogul, studiu de caz	
3. Tema 1: Specificarea uni mini-limbaj de programare si implementarea unui analizor lexical 1.3 utilizare lex	Discutarea datelor de test, evaluare	
4. Tema 1: Specificarea uni mini-limbaj de programare si implementarea unui analizor lexical 1.4 Detreminarea atomilor lexicali folosind automate finite	Explicatia, dialogul, studiu de caz	
5. Tema 2: Analiza sintactica: 2.1 Determinarea gramaticii pentru structuri sintactice specificate	Discutarea datelor de test, evaluare	
6. Tema 2: Analiza sintactica 2.2 Utilizarea yacc	Explicatia, dialogul, studiu de caz, discutarea datelor de test	
7. Prezentare tema 2	Discutarea datelor de test, evaluare	
<p>Bibliografie</p> <p>1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.</p> <p>2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973.</p> <p>3. D. GRIES - Compiler construction for digital computers,, John Wiley, New York, 1971.</p> <p>4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006</p> <p>5. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987.</p> <p>6. <a href="http://www.cs.ubbcluj.ro/~adriana/LFTC">http://www.cs.ubbcluj.ro/~adriana/LFTC</a></p>		

## 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	30%
10.5 Seminar			
10.6 Laborator	- sa fie capabili sa implementeze conceptele si algoritmi cursului - aplicarea tehnicilor prezentate pentru diferite clase de limbaje de programare	Verificare continua a activitatii in timpul orelor de laborator - verificarea documentatiei - verificarea programelor	70%

### 10.7 Standard minim de performanță

**Prezenta: minim 75% la activitatile de seminar (5 din 7 seminarii) si minim 90% la activitatile de laborator (6 din 7 laboratoare).** Studentii care nu indeplinesc criteriile de prezenta nu vor putea participa la examenul scris nici in sesiunea normala, nici in sesiunea de restante.

- Predarea laboratoarelor este posibila numai in timpul activitatilor didactice din timpul semestrului.
- Nu se vor putea preda laboratoare in sesiunea de restante.
- Studentii vor participa la seminarii cu grupa lor.
- Recuperarea seminariilor in alte intervale orare decat cele din orar se va face numai cu acordul cadrului didactic.
- Studentii se vor prezenta la laboratoare cu semigrupa in care sunt repartizati.
- **Recuperarea prezentelor** la laborator se poate face la orice cadru didactic care preda LFTC la specializarea MIR, **dar predarea laboratoarelor se va face numai la cadrul didactic cu care se desfasoara activitatea uzuala** a studentului (nu se vor nota laboratoare de catre alt cadru didactic).
- Motivarea absentelor se va face numai in baza unei adeverinte medicale parafate la cabinetul medical studentesc.

### Notare:

- Cel puțin nota **5 la examenul scris din sesiunea de examene/restante** desfasurat pe platforma moodle, cursul LFTC-MI, cu prezenta **fizica** a studentilor in sala si la ora stabilite in Academic Info
- cel puțin nota **6 pentru nota pt. Laborator**, calculata ca si medie aritmetica a notelor pe laboratoare. Un laborator nepredat va fi notat cu nota 1. Intarzierile in predarea laboratoarelor vor fi penalizate cu 1 punct/2 saptamani, dar nu mai mult de 4 saptamani. Dupa aceste 4 saptamani de la enuntarea temei, laboratorul se va considera nepredat, in absenta unei motivari medicale corespunzatoare perioadei de absente.

De aici rezulta si cel putin nota 5 pentru media calculata.

### **Bonusuri**

La nota obtinuta se pot adauga bonusuri din activitati de seminar si laborator, astfel:

- **seminar:**
  - 50 puncte pentru o iesire la tabla si rezolvarea completa si corecta a unui exercitiu fara sprijinul cadrului didactic
  - 30 puncte pentru rezolvarea individual in banca a unei probleme
  - Maxim 20 puncte pentru iesirea la tabla si rezolvarea unei probleme cu sprijin din partea cadrului didactic
  - Maxim 30 puncte pentru solutii alternative propuse sau abordari diferite si corecte pentru problemele propuse
  - Un total de 500 puncte bonus la seminar valoreaza 1 punct adaugat notei finale. Pentru punctaje sub 500 puncte se va calcula proportional ponderea bonusului.
- La **laborator** se poate obtine un bonus de pana la 0.5 puncte adaugat notei finale de laborator pentru corectitudine, validari, cod curat, review la codul colegilor, bonus acordat de catre cadrul didactic responsabil la laborator.

Data completării

29.09.2023

Semnătura titularului de curs

Conf. Dr. GURAN Adriana

Semnătura titularului de seminar

Conf. Dr. GURAN Adriana

Data avizării în departament

.....

Semnătura directorului de departament

.....