

FIȘA DISCIPLINEI

Limbaje formale și tehnici de compilare

Anul universitar 2026-2027

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 informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licenta
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Limbaje formale și tehnici de compilare			Codul disciplinei	MLR5023
2.2. Titularul activităților de curs	lect. dr. Lupșa Dana				
2.3. Titularul activităților de seminar	lect. dr. Lupșa Dana				
2.4. Anul de studiu	3	2.5. Semestrul	5	2.6. Tipul de evaluare	Examen
2.7. Regimul disciplinei	Obligativ	2.8. Tipul disciplinei	Disciplină fundamentală (DF)		

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	6	din care: 3.2. curs	2	3.3. seminar/ laborator/ proiect	2sem +2lab
3.4. Total ore din planul de învățământ	84	din care: 3.5. curs	28	3.6 seminar/laborator	56
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					6
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					14
Tutoriat (consiliere profesională)					5
Examinări					6
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				41	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	- structuri de date și algoritmi
4.2. de competențe	- abilitati de programare (nivel mediu)

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	sala dotată cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	laborator dotat cu calculatoare; mediu de dezvoltare pt. limbaje de programare de nivel înalt

6.1. Competențele dobândite în urma absolvirii programului de studii

Competențe profesionale	
Codul competenței	Competență
CP1	crează softuri
CP3	analizează specificații software
CP5	definește cerințe tehnice
Competențe transversale	
Codul competenței	Competență
CT1	Lucrează independent.
CT2	Soluționează probleme.
CT3	Gândește analitic

6.2. Rezultatele învățării specifice programului de studii

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP1 CP6	1. Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.	1. Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
CP5	2. Studentul/absolventul alege, explică și specifică fundamentele matematice aplicate în informatică, inclusiv logica formală, algebra, probabilitățile și statisticile.	2. Studentul/absolventul aplică, evaluează, propune metodele matematice pentru modelarea, simularea și rezolvarea problemelor informatice.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. cunoașterea, înțelegerea și folosirea conceptelor informatice teoretice de bază folosite în proiectarea compilatoarelor
2. îmbunătățirea abilităților de programare.
3. capacitatea de înțelegere a unor specificații formale
Abilități academice specifice (Specific academic skills)
1. ... 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)
2. înțelegerea conceptelor limbajelor formale și dezvoltarea abilităților de a modela diverse fenomene folosind limbaje formale ; abilitatea de a aplica tehnici specific compilatoarelor pentru diferite probleme din viața reală

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Limbaje formale, compilator: prezentare generala. Analiza lexicala	Expunere, descriere, explicatie, exemplificare	
2. Automate finite.	Expunere, descriere, explicatie, exemplificare	
3. Limbaje regulate: teoreme echivalenta, proprietati de inchidere, lema de pompare	Expunere, descriere, explicatie, exemplificare	
4. Expresii regulate. Generatoare de analizoare lexicale	Expunere, descriere, explicatie, exemplificare	
5. Limbaje independente de context. Proprietati.	Expunere, explicatie, exemplificare, studiu de caz	
6. Generatoare de analizoare sintactice. Automate Push Down	Expunere, explicatie, exemplificare, studiu de caz	
7,8,9,10. Analiza sintactica	Expunere, explicatie, exemplificare, studiu de caz	
11. Gramatici de atribute	Expunere, descriere, explicatie, exemplificare	
12. Cod intermediar . Modele de calcul	Expunere, descriere, explicatie, exemplificare	
13. Automate si transatoare. Masini Turing. Aplicatii si exemple	Expunere, descriere, explicatie, exemplificare, demonstrare	
14. Gramatici si automate. Exercitii, aplicatii si probleme recapitulative	Expunere: descrierea, exemplificarea, demonstrare	
<p>Bibliografie:</p> <ol style="list-style-type: none"> 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 Pub. Co., 1997. 6. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987. 		
8.2 Seminar / laborator	Metode de predare	Observații
Seminar		
1. Minilimbaje de programare: elemente lexicale si sintactice.	Explicatia, dialogul, studiu de caz	
2. Limbaje si specificari	Dialogul, dezbaterea, studiul de caz, exemplificarea	
3. Automate finite:	Dialogul, dezbaterea, studiul de caz, exemplificarea	
4,5. Limbaje regulate.	Dialogul, dezbaterea, studiul de caz, exemplificarea, demonstratia	
6. Expresii regulate. Legatura dintre automate finite si expresii regulate	Dialogul, studiul de caz, exemplificarea, demonstratia	
7. Gramatici de toate felurile	Dialogul, dezbaterea, studiul de caz, exemplificarea	
8. Automate push-down	Dialogul, dezbaterea, studiul de caz, exemplificarea,	
9,10,11,12. Analiza sintactica	Dialogul, dezbaterea, studiul de caz, exemplificarea	
13: Gramatici de atribute; cod intermediar	Dialogul, dezbaterea, studiul de caz, exemplificarea	
14. Traducere si transatoare. Alte probleme recapitulative.	Dialogul, dezbaterea, studiul de caz, exemplificarea	
Laborator		
1. Tema 1: Specificarea uni 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 uni mini-limbaj de programare si implementarea unui analizor lexical	Explicatia, dialogul, studiu de caz	

1.2 implemetarea functiilor principale ale analizorului		
3. Tema 1: Specificarea uni mini-limbaj de programare si implementarea unui analizor lexical 1.3 organizarea tabelii de simboluri	Explicatia, dialogul, studiu de caz	
4. Tema 1: Specificarea uni mini-limbaj de programare si implementarea unui analizor lexical 1.4 programul principal, testare 1.5 prezentare	Discutarea datelor de test, evaluare	
5. Tema 2: Automate finite 2.1 verificarea acceptarii unei secvente	Explicatia, dialogul, studiu de caz	
6. Tema 2: Automate finite: 2.2 Rescriere/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	
7. Tema 3: Implementarea unui analizor lexical folosind un generator de analizor lexical utilizare lex/flex	Explicatia, dialogul, studiu de caz, discutarea datelor de test, evaluare	
8. Tema 4: Implementarea unui analizor sintactic folosind un generator de analizor sintactic utilizare yacc/bison	Explicatia, dialogul, studiu de caz, discutarea datelor de test, evaluare	
9. Tema 5: Implementarea unui analizor sintactic 3.1 alegerea structurilor de date si arhitectura aplicatiei 3.2 implementarea functiilor principale ale analizorului	Explicatia, dialogul, studiu de caz	
10. Tema 5: Implementarea unui analizor sintactic 3.3 programul principal si integrarea modulelor	Explicatia, dialogul, studiu de caz	
11. Tema 5: Implementarea unui analizor sintactic 3.4 testare si corectarea posibilelor erori	Explicatia, dialogul, studiu de caz	
12. Tema 5: Implementarea unui analizor sintactic 3.5 prezentare	Discutarea datelor de test, evaluare	
13. Tema 6: Implementarea unui mini-compiler functional 6.1 specificare si implementare <ul style="list-style-type: none"> • utilizare lex/flex + yacc/bison • refolosire rezolvarea problemelor de la temele 3 si 4 generarea unui program asm echivalent semantic si testarea acestuia	Explicatia, dialogul, studiu de caz, discutarea datelor de test	
14. Tema6: Implementarea unui mini-compiler functional 6.2 testarea programului asm generat 6.3 prezentare	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. G. MOLDOVAN, V. CIOBAN, M. LUPEA - Limbaje formale si automate. Culegere de probleme, Univ. Babes-Bolyai, Cluj-Napoca, 1996		
5. MOTOGNA, S. - Metode de proiectare a compilatoarelor, Ed. Albastra, 2006		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	- cunoasterea principiilor de baza ale domeniului	Examen scris	60%
	- aplicarea conceptelor studiate la curs		
- rezolvarea de probleme			
9.5 Seminar/laborator	- sa fie capabili sa foloseasca 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 intelegerii conceptelor - verificarea programelor	40%
9.6 Standard minim de promovare			
Prezenta: minim 75% la activitatile de seminar si minim 90% la activitatile de laborator Notare: <ul style="list-style-type: none">• Cel putin nota 5 la examenul scris;• Cel putin nota 5 pentru nota laborator.			

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)

Nu se aplica

Data completării:

..12.05.2026.

Semnătura titularului de curs

Lect. Dr. Lupsa Dana

Semnătura titularului de seminar

Lect. Dr. Lupsa Dana.

Data avizării în departament:

...

Semnătura directorului de departament

Conf. dr. Adrian STERCA