

FIȘA DISCIPLINEI¹⁾

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Litere și Științe
1.3. Departamentul	Informatică, Tehnologia Informației, Matematică și Fizică
1.4. Domeniul de studii universitare	Informatică
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	Limbaje Formale și Compilatoare
2.2. Titularul activităților de curs	Lect. dr. Daniela Șchiopu
2.3. Titularul activităților seminar/laborator	Lect. dr. Daniela Șchiopu
2.4. Titularul activității proiect	-
2.5. Anul de studiu	II
2.6. Semestrul*	3
2.7. Tipul de evaluare	E
2.8. Categoria formativă** / regimul*** disciplinei	DF / O

*numărul semestrului este conform planului de învățământ;

**DF - Discipline fundamentale; DD - discipline de domeniu; DS - discipline de specialitate; DC - discipline complementare, DA - disciplina de aprofundare, DSI- disciplina de sinteza.

***obligatorie = O; opțională = A; facultativă = L

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	2	3.4. Proiect	0
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. Seminar/laborator	28	3.8. Proiect	0
3.9. Distribuția fondului de timp							ore
Studiu după manual, suport de curs, bibliografie și notițe							30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							30
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							28
Tutoriat							-
Examinări							6
Alte activități							
3.10 Total ore studiu individual	94						
3.11. Total ore pe semestru	150						
3.12. Numărul de credite	6						

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none">➤ Logică matematică și computațională➤ Fundamentele programării
--------------------	--------------------------------------------------------------------------------------------------------------------------

¹⁾ Adaptare după Ordinul Ministrului educației, cercetării, tineretului și sportului nr. 5 703/2011 privind implementarea Codului național al calificărilor din învățământul superior, publicat în Monitorul Oficial al României, partea I, nr.880 bis / 13.XII.2011

	➤ Metode avansate de programare
4.2. de competențe	<ul style="list-style-type: none"> ➤ Cunoașterea programării procedurale (paradigmă, limbaj, fundamente și elemente avansate); ➤ Cunoașterea unui limbaj de programare și a unui mediu de dezvoltare de programe în acel limbaj (de exemplu C, C++ etc.).

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	➤ sală de curs multimedia cu videoproiector și conexiune la Internet
5.2. de desfășurare a seminarului/laboratorului	➤ sală de laborator care să ofere software corespunzător pentru dezvoltarea de programe, precum și pentru generarea automată a componentelor unui compilator

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> ➤ C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date. ➤ C1.4 Testarea unor aplicații pe baza unor planuri de test. ➤ C2.3 Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice. ➤ C3.1 Descrierea de concepte, teorii și modele folosite în domeniul de aplicare. ➤ C3.2 Identificarea și explicarea modelelor informatice de bază adecvate domeniului de aplicare. ➤ C3.3 Utilizarea modelelor și instrumentelor informatice și matematice pentru rezolvarea problemelor specifice domeniului de aplicare. ➤ C3.4 Analiza datelor și a modelelor. ➤ C4.2 Interpretarea de modele matematice și informatice (formale). ➤ C4.3 Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale. ➤ C4.5 Încorporarea de modele formale în aplicații specifice din diverse domenii.
Competențe transversale	<ul style="list-style-type: none"> ➤ CT1. Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic, științific și profesional, în vederea valorificării creative a propriului potențial, cu respectarea principiilor și normelor de etică profesională; ➤ CT2. Desfășurarea eficientă a activităților organizate în echipă și dezvoltarea capacităților empatică și de comunicare inter-personală, de relaționare și colaborare cu persoane și grupuri diverse implicate în dezvoltarea și utilizarea de sisteme software; ➤ CT3. Utilizarea de metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, dar și de adaptare la cerințele unei societăți dinamice și în continuă schimbare, precum și dezvoltarea capacității de a comunica eficient și profesionist atât în limba română, cât și într-o limbă de circulație internațională, prin însușirea și folosirea adecvată a limbajului de specialitate.

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

7.1. Obiectivul general al disciplinei	➤ dobândirea de către studenți a cunoștințelor necesare pentru a înțelege utilitatea și funcționalitatea compilatoarelor, dar și a modului de proiectare și dezvoltare a acestora (bazat pe teoria limbajelor formale și a automatelor), precum și aplicarea acestor
----------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	cunoștințe în rezolvarea altor probleme înrudite.
7.2. Obiectivele specifice	<p>După parcurgerea disciplinei, studenții vor putea să:</p> <ul style="list-style-type: none"> ➤ Identifice și să descrie corect principalele componente și funcții ale unui compilator; ➤ Definească în mod corespunzător noțiuni variate de la gramatici și limbaje generate de acestea, la expresii regulate și automate de acceptare (recunoaștere); ➤ Rezolve corect aplicații cu gramatici și limbaje generate de către acestea, mulțimi, expresii și limbaje regulate, precum și cu automate finite deterministe și nedeterministe; ➤ Rezume corect etapele realizării unui compilator și să explice concis tehnicile specifice folosite la dezvoltarea fiecărei componente a acestuia; ➤ Proiecteze și să implementeze analizoare lexicale și sintactice de complexitate elementară; ➤ Utilizeze generatoare automate pentru realizarea componentelor unui compilator; <p>Interpreteze corect comportamentul compilatoarelor în interacțiune cu utilizatorii și cu programele lor.</p>

8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1. Introducere în compilare. Procesoare de limbaje. Structura și fazele unui compilator. Analiza lexicală, analiza sintactică, analiza semantică, generarea codului intermediar, optimizare cod, generare cod, managementul tabelii de simboluri, medii de construcție pentru componentele unui compilator.	2	prelegeri active și angajante, supervizare și mentorat "deschise", învățarea prin descoperire, învățare pe grupuri, învățare bazată pe rezolvarea de probleme, învățare centrată pe student, learning by doing, învățare hibridă, folosirea resurselor educaționale open disponibile online, învățare reflectivă etc.	
2. Un translator simplu, orientat de sintaxă. Definierea sintaxei (gramatici, derivări, arbori de sintaxă etc.), analiza lexicală, tabela de simboluri, generarea de cod intermediar.	2		
3. Introducere în teoria compilării și elemente de teoria limbajelor formale. Noțiunea de limbaj. Gramatici și limbaje Chomsky. Mulțimi regulate, expresii regulate, limbaje regulate. Automate de acceptare – automate finite deterministe și nedeterministe. Gramatici regulate. Expresii regulate. Automate push-down. Gramatici independente de context. Traducerea limbajelor.	12		
4. Proiectarea compilatoarelor. Analiza	10		

lexicală. Analiza sintactică - analiza descendentă (cu revenire, fără revenire) și analiza ascendentă (LR, SLR, canonică LR, LALR). Analiza semantică – definiții orientate de sintaxă, arbori de sintaxă, scheme de translatore orientată pe sintaxă.			
5. Generarea codului. Producerea codului intermediar. Medii de execuție. Generarea codului obiect. Optimizarea codului obiect.	2		

Bibliografie

- Aho, A. V., Sethi, R., Lam, M., Ullman, J. D., Compilers Principles, Techniques, and Tools, Addison Wesley, 2007.
- Aho, A. V., Ullman, J. D., The Theory of Parsing, Translation and Compiling, Prentice-Hall, 1973.
- Athanasiu, I., Limbaje formale și automate, Matrix Rom, București, 2002.
- Serbănati, L. D., Limbaje de programare și compilatoare, Editura Academiei, Bucuresti, 1987.
- Căzănescu V., Introducere în teoria limbajelor formale, Editura Academiei, Bucuresti, 1983.
- Marcus, S., Gramatici și automate finite, Editura Academiei, Bucuresti, 1964.
- Salomaa, A., Formal languages, Academic Press, New York, 1973.
- Resurse educaționale disponibile la:
Cursul Compilers - Stanford: <https://web.stanford.edu/class/cs143/>

8.2. Laborator	Nr. ore	Metode de predare	Observații
Aplicații cu gramatici și limbaje generate de către acestea, cu mulțimi, expresii și limbaje regulate, precum și cu automate finite deterministe și nedeterministe.	14	învățarea prin descoperire, învățare pe grupuri, învățare bazată pe rezolvarea de probleme, învățare centrată pe student, learning by doing, învățare hibridă, folosirea resurselor educaționale open disponibile online, învățare reflectivă	
Organizarea și proiectarea unui compilator.	10		
Generarea automată a componentelor unui compilator.	4		

Bibliografie

- Aho, A. V., Sethi, R., Lam, M., Ullman, J. D., Compilers Principles, Techniques, and Tools, Addison Wesley, 2007.
- Aho, A. V., Ullman, J. D., The Theory of Parsing, Translation and Compiling, Prentice-Hall, 1973.
- Athanasiu, I., Limbaje formale și automate, Matrix Rom, București, 2002.
- Serbănati, L. D., Limbaje de programare și compilatoare, Editura Academiei, Bucuresti, 1987.
- Căzănescu V., Introducere în teoria limbajelor formale, Editura Academiei, Bucuresti, 1983.
- Resurse educaționale disponibile la:
Cursul Compilers - Stanford: <https://web.stanford.edu/class/cs143/>

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

➤ Conținuturile disciplinei corespund cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului, așa după cum rezultă din prezenta fișă, dar și din fișa specializării, acestea fiind în concordanță deplină cu CNCIS și COR.	➤
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Dobândirea competențelor profesionale și transversale specifice disciplinei	<i>Evaluare scrisă:</i> examinarea finală (lucrare scrisă, cu subiecte teoretice și aplicații)	40% N1= nota la această probă este de la 1 la 10 (1 punct din oficiu)
10.5. Laborator		<i>Evaluare laborator:</i> teme de laborator, testare scrisă, dezvoltarea de programe pentru implementarea algoritmilor specifici, activitatea la laborator	60% N2= nota la această probă este de la 1 la 10 (1 punct din oficiu)
Nota finală este: $40\% \cdot N1 + 60\% \cdot N2$			
10.6. Standard minim de performanță			
<ul style="list-style-type: none"> ➤ RNCIS: Implementarea și documentarea de unități de program în limbaje de programare de nivel înalt și folosirea eficientă a mediilor de programare; Modelarea și rezolvarea unor probleme cu grad redus de complexitate, folosind cunoștințe de matematică și informatică; Realizarea și întreținerea unor aplicații informatice pentru rezolvarea unor probleme reale de complexitate redusă; Realizarea componentelor informatice pentru o aplicație dedicată de complexitate medie; ➤ <i>Dezvoltarea de programe cu un grad redus de dificultate pentru manipularea gramaticilor și a limbajelor generate de acestea, pe baza algoritmilor specifici; utilizarea generatoarelor automate pentru componentele unui compilator.</i> 			

Data
completării
21.09.2020

Semnătura titularului de curs
Lect. dr. Daniela Șchiopu

Semnătura titularului de laborator
Lect. dr. Daniela Șchiopu

Data avizării în departament
21.09.2020

Semnătura directorului de departament
Conf. dr. Gabriela Moise