

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	Tehnici avansate de programare
2.2. Titularul activităților de curs	Conf. dr. Simona Nicoară
2.3. Titularul activităților aplicative	Conf. dr. Simona Nicoară
2.4. Anul de studiu	II
2.5. Semestrul*	4
2.6. Tipul de evaluare	E, scris
2.7. Categoria formativă** / regimul*** disciplinei	S2/ O

*numărul semestrului este conform planului de învățământ;

** fundamentală = F0; de domeniu = D1; de specialitate = S2; complementară = C3

***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. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6. Seminar/laborator	28
3.7. Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					25
Tutoriat					5
Examinări					4
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	Fundamentele programării, Programare orientată pe obiecte Algoritmi fundamentali, Structuri de date Metode avansate de programare
4.2. de competențe	Programarea calculatoarelor

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

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	➤ sală de curs dotată cu videoprojector ➤ suport electronic la https://timf.upg-ploiesti.ro/cursuri
5.2.	➤ tehnica de calcul

6. Competențe specifice acumulate

Competențe profesionale	<p>C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstracție (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</p> <p>C2.1. Identificarea de metodologii adecvate de dezvoltare a sistemelor software</p> <p>C2.2. Identificarea și explicarea mecanismelor adecvate de specificare a sistemelor Software</p> <p>C2.3 Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice</p> <p>C2.4 Utilizarea de criterii și metode adecvate pentru evaluarea aplicațiilor informatice</p> <p>C2.5 Realizarea unor proiecte informatice dedicate</p> <p>C3.2 Identificarea și explicarea modelelor informatice de bază adecvate domeniului de aplicare</p> <p>C3.3 Utilizarea modelelor și instrumentelor informatice și matematice pentru rezolvarea problemelor specifice domeniului de aplicare</p>
Competențe transversale	<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</p>

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

7.1. Obiectivul general al disciplinei	Familiarizarea studenților cu tehnicile avansate de programare în scopul dezvoltării abilității de alegere adecvate a metodelor de rezolvare a problemelor din lumea reală
7.2. Obiectivele specifice	<p>La sfârșitul cursului, studentul va fi capabil să:</p> <ul style="list-style-type: none"> • descrie conceptele fundamentale ale domeniului, numească și descrie tehnicile de programare clasice (cunoștințe dobândite) • recunoască cele mai adecvate tehnici de rezolvare pentru probleme particulare, explice funcționarea fiecărei tehnici, înțeleagă modul de determinare a ordinului de complexitate a unui algoritm (înțelegere) • implementeze și testeze tehnicile învățate (aplicare) • analizeze comparativ tehnicile de programare, inclusiv după complexitate (analiză)

8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1. Vedere de ansamblu asupra cursului. Prezentarea modului de evaluare. Recapitulare algoritmi și metode avansate de programare	2	<ul style="list-style-type: none"> • predare interactivă • învățare prin descoperire • problematizare 	
2. Elemente de analiză a complexității algoritmilor. Exemple. Analiza algoritmilor de sortare și de căutare. Clase de complexitate	4		
3. Tehnici clasice de programare: tehnica forței brute, tehnica reducerii dimensiunii problemei	2		Sinteza cursului se afla

(recursivitatea ca tehnica de programare), tehnica Greedy, tehnica Programarii dinamice, tehnica Backtracking, tehnica Branch and Bound, tehnica Divide et Impera	4 4 4 4 2	online la dispozitia studentilor (https://timf.upg-ploiesti.ro/cursuri)	
4. Vedere de ansamblu asupra tehnicilor avansate de programare	2		
Bibliografie			
1. Valentin Cristea, Irina Athanasiu, Eugenia Kalisz, <i>Tehnici de programare</i> , Ed. Teora, Bucuresti, 1996			
2. Donald E. Knuth, <i>The art of computer programming, Vol. I – Fundamental Algorithms</i> , ediția a treia, Addison Wesley Longman, 1997			
3. Donald E. Knuth, <i>The art of computer programming, Vol. III – Sorting and Searching</i> , ediția a doua, Addison Wesley Longman, 1998			
4. Livovschi L., Georgescu H. <i>Sinteza și analiza algoritmilor</i> , Universitatea din București, Fac. de Matematică, București, 1985			
Cormen, T.H., Leiserson, C.E., Rivest R.R., Stein C., <i>Introduction to algorithms</i> , MIT, 3rd ed, 2009			
5. Sedgewick, R., Wayne, K., <i>Algorithms</i> , 4th ed., Addison-Wesley, 2011			
6. Skiena, S., <i>The algorithm design manual</i> , 2nd ed., Springer Verlag, London, 2010			
7. Tim Roughgarden, <i>Algorithms: design and analysis</i> , Part I and II, Stanford University, Coursera, 2015			
8.2. Laborator	Nr. ore	Metode de predare	Observații
1. Prezentarea modului de desfasurare a laboratorului. Recapitulare metode avansate de programare	2	<ul style="list-style-type: none"> • exercițiul • problematizarea • invatarea prin descoperire 	
2. Analiza complexitatii algoritmilor	2		
3. Tehnica Greedy. Aplicatii	4		
4. Tehnica Backtracking. Aplicatii	4		
5. <i>Verificare de laborator</i>	2		Probă practică
6. Tehnica Programarii dinamice. Aplicatii	4		
7. Tehnica Branch and Bound. Aplicatii	4		
8. Tehnica Divide et Impera. Aplicatii	2		
9. Rezolvarea unor probleme cu mai multe tehnici de programare adecvate. Analiza comparativa	2		
10. <i>Verificare de laborator</i>	2		Probă practică
Bibliografie			
1. Valentin Cristea, Irina Athanasiu, Eugenia Kalisz, <i>Tehnici de programare</i> , Ed. Teora, Bucuresti, 1996			
2. Skiena, S., <i>The algorithm design manual</i> , 2nd ed., Springer Verlag, London, 2010			
3. Tim Roughgarden, <i>Algorithms: design and analysis</i> , Part I and II, Stanford University, Coursera, 2015			

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

➤ Aprofundarea teoretică și practică a disciplinei este o condiție de bază pentru dezvoltarea și întreținerea aplicațiilor informatice destinate rezolvării problemelor din context real. Cursul și laboratorul

sunt astfel concepute încât, prin competențele formate, să răspundă cerințelor pieței muncii.

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"> calitatea răspunsurilor la examen, coerența argumentării, calitatea corelațiilor efectuate completitudinea și corectitudinea cunoștințelor; capacitatea de a opera cu cunoștințele asimilate; gradul de asimilare a limbajului de specialitate și capacitatea de comunicare; 	Lucrare scrisă cu subiecte teoretice și aplicative	40% Pentru promovarea examenului este necesară minim nota 5
10.5 Laborator	<ul style="list-style-type: none"> capacitatea de aplicare în practică, în contexte diferite, a cunoștințelor învățate; capacitatea de a opera cu cunoștințele asimilate; capacitatea de analiză, de interpretare personală; 	Evaluare sumativă - probă practică: realizarea de aplicații care folosesc tehnici avansate de programare	50% Pentru promovarea examenului este necesară minim nota 5
		Din oficiu	10%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> Asimilarea limbajului de specialitate Abilitatea de selectare a tehnicilor adecvate de rezolvare pentru o problema dată 			

Data completării
17 sept. 2018

Semnătura titularului de curs
Conf. dr. Simona Nicoară

Semnătura titularului de laborator
Conf. dr. Simona Nicoară

Data avizării în departament

Semnătura directorului de departament
Conf. dr. Gabriela Moise