

# FIȘA DISCIPLINEI<sup>1)</sup>

## 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 seminar/laborator	Conf. dr. Simona Nicoară, lector dr. Elia Dragomir
2.4. Titularul activității proiect	-
2.5. Anul de studiu	II
2.6. Semestrul*	4
2.7. Tipul de evaluare	Examen
2.8. Categoria formativă** / regimul*** disciplinei	DS / 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. laborator	2	3.4. Proiect	-
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. laborator	28	3.8. Proiect	-
3.9. 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							-
3.10 Total ore studiu individual	69						
3.11. Total ore pe semestru	125						
3.12. Numărul de credite	5						

## 4. Precondiții

4.1. de curriculum	<ul style="list-style-type: none"><li>➤ Algoritmi fundamentali</li><li>➤ Fundamentele programării</li></ul>
--------------------	---

<sup>1)</sup> 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

	<ul style="list-style-type: none"> <li>➤ Structuri de date</li> <li>➤ Metode avansate de programare</li> </ul>
4.2. de competențe	➤ Programarea calculatoarelor

## 5. Condiții

5.1. de desfășurare a cursului	➤ suport electronic la <a href="https://timf.upg-ploiesti.ro/cursuri/">https://timf.upg-ploiesti.ro/cursuri/</a> TAP
5.2. de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>➤ tehnica de calcul</li> <li>➤ mediu de programare adecvat</li> </ul>

## 6. Competențe specifice acumulate

<b>Competențe profesionale</b>	<p>C1. PROGRAMAREA ÎN LIMBAJE DE NIVEL ÎNALT</p> <ul style="list-style-type: none"> <li>• C1.2 EXPLICAREA UNOR APLICAȚII SOFTWARE EXISTENTE, PE NIVELURI DE ABSTRACȚARE (ARHITECTURĂ, PACHETE, CLASE, METODE) UTILIZAND IN MOD ADECVAT CUNOȘTINȚELE DE BAZĂ</li> <li>• C1.3 ELABORAREA CODURILOR SURSĂ ADECVATE ȘI TESTAREA UNITARĂ A UNOR COMPONENTE INTR-UN LIMBAJ DE PROGRAMARE CUNOSCUT, PE BAZA UNOR SPECIFICAȚII</li> <li>• C1.4 TESTAREA UNOR APLICAȚII PE BAZA UNOR PLANURI DE TEST</li> <li>• C1.5 DEZVOLTAREA DE UNITĂȚI DE PROGRAM ȘI ELABORAREA DOCUMENTAȚIILOR AFERENTE</li> </ul> <p>C2. DEZVOLTAREA ȘI ÎNTREȚINEREA APLICAȚIILOR INFORMATICE</p> <ul style="list-style-type: none"> <li>• C2.1. IDENTIFICAREA DE METODOLOGII ADECVATE DE DEZVOLTARE A SISTEMELOR SOFTWARE</li> <li>• C2.2. IDENTIFICAREA ȘI EXPLICAREA MECANISMELOR ADECVATE DE SPECIFICARE A SISTEMELOR SOFTWARE</li> <li>• C2.3 UTILIZAREA METODOLOGIILOR, MECANISMELOR DE SPECIFICARE ȘI A MEDIILOR DE DEZVOLTARE PENTRU REALIZAREA APLICAȚIILOR INFORMATICE</li> <li>• C2.4 UTILIZAREA DE CRITERII ȘI METODE ADECVATE PENTRU EVALUAREA APLICAȚIILOR INFORMATICE</li> <li>• C2.5 REALIZAREA UNOR PROIECTE INFORMATICE DEDICATE</li> </ul> <p>C3. UTILIZAREA INSTRUMENTELOR INFORMATICE IN CONTEXT INTERDISCIPLINAR</p> <ul style="list-style-type: none"> <li>• C3.1 DESCRIEREA DE CONCEPTE, TEORII SI MODELE FOLOSITE IN DOMENIUL DE APLICARE.</li> <li>• C3.2 IDENTIFICAREA SI EXPLICAREA MODELELOR INFORMATICE DE BAZA ADECVATE DOMENIULUI DE APLICARE</li> <li>• C3.3 UTILIZAREA MODELELOR SI INSTRUMENTELOR INFORMATICE SI MATEMATICE PENTRU REZOLVAREA PROBLEMELOR SPECIFICE DOMENIULUI DE APLICARE</li> </ul>
<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 acunoștințelor, 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 specifice acumulate)

7.1. Obiectivul general al disciplinei	Familiarizarea studenților cu strategiile algoritmice (tehnicile avansate de programare) in scopul dezvoltarii abilitatii de alegere adecvata 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"> <li>• descrie conceptele fundamentale ale domeniului, numească și descrie strategiile algoritmice de programare prezentate (cunoștințe dobândite)</li> <li>• identifice exemple practice la care se aplica tehnicile prezentate; recunoască</li> </ul>

	<p>cele mai adecvate tehnici de rezolvare pentru probleme particulare; explice funcționarea fiecărei tehnici; intelega modul de determinare a ordinului de complexitate a unui algoritm; formuleze un spatiu de probleme eficient pentru o problema data; descrie explozia combinatoriala a spatiului de cautare si consecintele sale; formuleze o problema ca CSP si sa o rezolve (înțelegere)</p> <ul style="list-style-type: none"> <li>• selecteze, implementeze si testeze tehnicile adecvate (aplicare)</li> <li>• analizeze comparativ tehnicile de programare, inclusiv dupa complexitate, descrie diferentele dintre brute-force si metodele euristice; compare elementele strategiilor de cautare cu cele ale jocurilor (analiză)</li> </ul>
--	--

## 8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1. Vedere de ansamblu asupra cursului. Recapituare analiza complexitatii algoritmilor, brute force, tehnica reducerii dimensiunii problemei, backtracking, divide et impera	4	<ul style="list-style-type: none"> <li>• predare interactivă</li> <li>• invatare prin descoperire</li> <li>• problematizare</li> </ul> <p>Sinteza cursului se afla online la dispozitia studentilor (<a href="https://timf.upg-ploiesti.ro/cursuri/TAP">https://timf.upg-ploiesti.ro/cursuri/TAP</a>)</p>	
2. Algoritmi greedy. Analiza optimalitatii solutiei	2		
3. Programare dinamica	4		
4. Metode euristice de rezolvare a problemelor. Branch and Bound	2		
5. Cautare combinatoriala. Probleme de satisfacere a restrictiilor (CSP – Constraint Satisfaction Problems). Backtracking si cautare locala	4		
6. Spatii de probleme. Probleme de cautare. Rezolvarea problemelor prin cautare: căutare pe nivel, căutare de cost uniform, căutare în adâncime (limitată), căutare în adâncime cu nivel iterativ, cautare bidirecțională. Eficienta cautarii	8		
7. Vedere de ansamblu asupra tehnicilor avansate de programare. Analiza comparativa. Determinarea tehnicii adecvate pt o problema data	4		

### Bibliografie

1. Donald E. Knuth, *The art of computer programming, Vol. I – Fundamental Algorithms*, ediția a treia, Addison Wesley Longman, 1997
2. Donald E. Knuth, *The art of computer programming, Vol. III – Sorting and Searching*, ediția a doua, Addison Wesley Longman, 1998
3. Valentin Cristea, Irina Athanasiu, Eugenia Kalisz, *Tehnici de programare*, Ed. Teora, Bucuresti, 1996
4. Livovschi L., Georgescu H. *Sinteza și analiza algoritmilor*, Universitatea din București, Fac. de Matematică, București, 1985
5. Cormen, T.H., Leiserson, C.E., Rivest R.R., Stein C., *Introduction to algorithms*, MIT, 3rd ed, 2009
6. Sedgewick, R., Wayne, K., *Algorithms*, 4th ed., Addison-Wesley, 2011
7. Skiena, S., *The algorithm design manual*, 2nd ed., Springer Verlag, London, 2010
7. Tim Roughgarden, *Algorithms: design and analysis, Part I and II*, Stanford University, Coursera, 2015

8. Russell, S., Norvig, P., <i>Artificial Intelligence: A Modern Approach</i> , 3rd ed., Pearson Education Inc., Prentice Hall, New Jersey, 2014			
<b>8.2. Laborator</b>	Nr. ore	Metode de predare	Observații
1. Recapitulare metode avansate de programare, analiza complexitatii algoritmilor, recursivitate, brute force, tehnica reducerii dimensiunii problemei	2	<ul style="list-style-type: none"> <li>• exercițiul</li> <li>• problematizarea</li> <li>• învățarea prin descoperire</li> </ul>	Inclusa analiza complexitatii computationale
2. Recapitulare Divide et Impera, Backtracking	2		
3. Tehnica Greedy. Aplicatii. Determinarea daca regula greedy aleasa conduce la o solutie optima	4		
4. Tehnica Programarii dinamice. Aplicatii	4		
5. Tehnica Branch and Bound. Aplicatii. Descrierea modului in care branch and bound poate imbunatati performanta unei metode euristice	4		
6. <i>Verificare de laborator</i> - Probă practică	2		
7. Căutare pe nivel, căutare de cost uniform, căutare în adâncime (limitată), căutare în adâncime cu nivel iterativ, cautare bidirecțională Proiect individual - <i>Verificare de laborator</i>	8		
8. Sinteze laborator	2		
Bibliografie			
<ol style="list-style-type: none"> <li>1. Valentin Cristea, Irina Athanasiu, Eugenia Kalisz, <i>Tehnici de programare</i>, Ed. Teora, Bucuresti, 1996</li> <li>2. Skiena, S., <i>The algorithm design manual</i>, 2nd ed., Springer Verlag, London, 2010</li> <li>3. Tim Roughgarden, <i>Algorithms: design and analysis</i>, Part I and II, Stanford University, Coursera</li> </ol>			
<b>8.3. Proiect</b>	Nr. ore	Metode de predare	Observații
Bibliografie			

### 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"> <li>• calitatea răspunsurilor la examen, coerența argumentării, calitatea corelațiilor efectuate</li> <li>• completitudinea și corectitudinea cunoștințelor;</li> <li>• capacitatea de a opera cu cunoștințele asimilate;</li> <li>• gradul de asimilare a limbajului de specialitate și capacitatea de comunicare;</li> </ul>	Lucrare scrisă cu subiecte teoretice si aplicative	40%
		Din oficiu	10%
10.5. Seminar/laborator	<ul style="list-style-type: none"> <li>• capacitatea de aplicare în practică, în contexte diferite, a cunoștințelor învățate;</li> <li>• capacitatea de analiză, de interpretare personală;</li> </ul>	Evaluare sumativă - probă practică: realizarea de aplicații care folosesc tehnici avansate de programare	50%
10.7. Standard minim de performanță			
<ul style="list-style-type: none"> <li>➤ Asimilarea limbajului de specialitate</li> <li>➤ Selectarea tehnicilor adecvate de rezolvare pentru o problema reala data, implementarea și documentarea adecvată de componente informatice pentru aplicații dedicate acestui scop</li> </ul>			

Data  
completării  
20 sept. 2022

Semnătura titularului de  
curs

Semnătura titularului de laborator

Data avizării în  
departament  
21 sept. 2022

Director de departament  
Conf. dr. Gabriela Moise

Decan  
Prof. univ. dr. Mihaela Suditu