

# 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	Algoritmi fundamentali
2.2. Titularul activităților de curs	Conf. dr. Simona Nicoară
2.3. Titularul activităților seminar/laborator	Conf. dr. Simona Nicoară, Inf. Isabela Hasnas
2.4. Titularul activității proiect	-
2.5. Anul de studiu	I
2.6. Semestrul*	1
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. 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							36
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							30
Tutoriat							4
Examinări							4
Alte activități							-
3.10 Total ore studiu individual	94						
3.11. Total ore pe semestru	150						
3.12. Numărul de credite	6						

## 4. Precondiții

4.1. de curriculum	➤
4.2. de competențe	➤

<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

## 5. Condiții

5.1. de desfășurare a cursului	➤ suport electronic la <a href="https://timf.upg-ploiesti.ro/cursuri/AF">https://timf.upg-ploiesti.ro/cursuri/AF</a>
5.2. de desfășurare a seminarului/laboratorului	➤ tehnica de calcul ➤ mediu de programare CodeBlocks

## 6. Competențe specifice acumulate

<b>Competențe profesionale</b>	<p>C1. PROGRAMAREA ÎN LIMBAJE DE NIVEL ÎNALT</p> <ul style="list-style-type: none"> <li>• C1.1 DESCRIEREA ADECVATĂ A PARADIGMELOR DE PROGRAMARE ȘI A MECANISMELOR DE LIMBAJ SPECIFICE, PRECUM ȘI IDENTIFICAREA DIFERENȚEI DINTRE ASPECTELE DE ORDIN SEMANTIC ȘI SINTACTIC</li> <li>• C1.2 EXPLICAREA UNOR APLICAȚII SOFTWARE EXISTENTE, PE NIVELURI DE ABSTRACȚARE (ARHITECTURĂ, PACHETE, CLASE, METODE) UTILIZAND ÎN 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>C4. UTILIZAREA BAZELOR TEORETICE ALE INFORMATICII SI A MODELELOR FORMALE</p> <ul style="list-style-type: none"> <li>• C4.1 DEFINIREA CONCEPTELOR ȘI PRINCIPIILOR DE BAZĂ ALE INFORMATICII, PRECUM ȘI A TEORIILOR ȘI MODELELOR MATEMATICE</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 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 specifice acumulate)

7.1. Obiectivul general al disciplinei	Familiarizarea studenților cu algoritmica și rolul său în rezolvarea 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 algoritmicii și proprietățile necesare ale unui algoritm, numească și descrie structurile algoritmice fundamentale și etapele rezolvării problemelor cu calculatorul, descrie principalele metode algoritmice de selecție, căutare și sortare a datelor, descrie subrutinele și recursivitatea, cunoască principiile de proiectare a algoritmilor, cunoască stilurile de lucru în proiectarea algoritmilor, cunoască aplicații practice ale algoritmilor în diverse domenii, își formeze o idee introductivă despre algoritmii neconvenționali (cunoștințe dobândite)</li> <li>• deceleze apelarea la algoritmi în viața de zi cu zi, formuleze corect o problemă, dezvolte gândirea algoritmică, recunoască cei mai adecvați algoritmi pentru probleme particulare, explice funcționarea fiecărei structuri algoritmice fundamentale, înțeleagă comportamentul unui algoritm, deosebească problemele de instanțe de probleme, testeze algoritmi cu date de test acoperitoare, înțeleagă rolul subrutinelor, compare metodele de căutare și de sortare, înțeleagă relația algoritmică între operația de sortare și operația de căutare, evalueze importanța algoritmilor în rezolvarea de probleme, dezbată cum poate fi rezolvată o problemă prin mai mulți algoritmi diferiți (înțelegere)</li> <li>• creeze algoritmi pentru rezolvarea de probleme de complexitate redusă-medie, folosească un limbaj de programare pentru a implementa, testa și depana algoritmi</li> </ul>

	<p>(aplicare)</p> <ul style="list-style-type: none"> <li>analizeze comparativ structurile decizionale, structurile iterative și algoritmi care rezolvă o aceeași problemă, aleagă cel mai potrivit algoritm într-o situație dată, analizeze cazurile particulare, identifice dacă un algoritm este corect sau nu, identifice proprietățile datelor și /sau altor condiții care conduc la comportamente diferite ale algoritmilor, execute algoritmi cu intrări diferite ca dimensiune și compară performanța lor, determine dacă pentru o problemă dată este adecvată o soluție recursivă sau iterativă (analiză)</li> </ul>
--	--

## 8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1. De ce se numește cursul Algoritmi Fundamentali? Ce este un algoritm? Proprietati Relatia computing - rezolvarea de probleme - algoritmi. Algoritmii in viata noastra. Gandire algoritmica Probleme si instante Aplicații practice ale algoritmilor	3	<ul style="list-style-type: none"> <li>predare interactivă folosind numeroase exemple</li> <li>învățare prin descoperire</li> </ul> <p>Sinteza cursului se afla online la dispozitia studentilor (<a href="https://timf.upg-ploiesti.ro/cursuri">https://timf.upg-ploiesti.ro/cursuri</a>)</p>	Formularea unor probleme din lumea reala
2. Descrierea algoritmilor. Teorema de structura. Structuri algoritmice fundamentale (structuri de control): secventa, decizia, iteratia	2		
3. Preliminarii matematice: numere, puteri, sume si produse, permutari si factoriale, numere armonice, numere Fibonacci, numere perfecte, inductie matematica	4		
4. Rezolvarea de probleme cu algoritmi. Etape. Formalizarea si modelarea lumii reale. Tipuri de probleme care pot fi rezolvate folosind calculatorul	2		
5. Principii de proiectare a algoritmilor (abstractizare, descompunere; brute force, transform & conquer etc.). Stiluri si scheme de lucru. Algoritmi corecti, algoritmi incorecti	2		
6. Subrutine (implementare ca proceduri si functii) Modularitate Recursivitate Algoritmi iterativi vs. algoritmi recursivi	2		
7. Selectie, cautare, sortare – prezentare introductiva. Cautare secventiala, cautare secventiala rapida, cautare secventiala in tabel ordonat, cautare prin interpolare. Relatia sortare-cautare	4		
8. Algoritmii in nucleul tehnologiei. Eficienta algoritmilor – introducere. Masurari empirice ale eficientei. Comportamentul unui algoritm in cel mai favorabil, cel mai nefavorabil caz si in cazul mediu. Notatia asimptotica. Clase de complexitate. Probleme intractabile	3		
9. Alte abordari in algoritmica (introducere): algoritmi de aproximare (pentru probleme intractabile), algoritmi euristici, randomized algorithms, algoritmi inspirati din biologie, algoritmi cuantici	4		
10. Resurse algoritmice (colectii de algoritmi, colectii de programe, surse de date etc.) si platforme de	2		

concursuri			
Bibliografie			
1. Donald E. Knuth, <i>The art of computer programming, Vol. I – Fundamental Algorithms</i> , ediția a treia, Addison Wesley Longman, 1997			
2. Donald E. Knuth, <i>The art of computer programming, Vol. III – Sorting and Searching</i> , ediția a doua, Addison Wesley Longman, 1998			
3. *Cormen, T.H., Leiserson, C.E., Rivest R.R., Stein C., <i>Introduction to algorithms</i> , MIT, 3rd ed, 2009			
4. Cristian Marinoiu, Simona Nicoara, <i>Introducere in informatica</i> , Ed. UPG, Ploiesti, 2014			
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			
<b>8.2. Laborator</b>	Nr. ore	Metode de predare	Observații
1. Prezentarea modului de desfasurare a laboratorului Discutii libere despre algoritmi. Cum poate fi rezolvata o problema prin mai multi algoritmi	2	<ul style="list-style-type: none"> <li>• exercițiul</li> <li>• problematizarea</li> <li>• învățare prin descoperire</li> </ul> folosirea instrumentelor de depanare a programelor	
2. Secvente si decizii (pseudocod/schema logica structurata si implementare in limbajul C)	4		
3. Iteratii (pseudocod/schema logica structurata si implementare in limbajul C). Analiza comparativa	6		
4. <i>Verificare de laborator 1 – abilitati de rezolvare de probleme cu algoritmi</i>	2		Probă practică
5. Vectori, matrici. Selectie, cautare	6		
6. Functii. Recursivitate	4		
7. Bubble sort, sortare prin selectie	2		
8. <i>Verificare de laborator 2</i>	2		Probă practică
Bibliografie			
Cristian Marinoiu, Simona Nicoara, <i>Introducere in informatica</i> , Ed. UPG, Ploiesti, 2014			
Donald E. Knuth, <i>The art of computer programming, Vol. I – Fundamental Algorithms</i> , ediția a treia, Addison Wesley Longman, 1997			
Donald E. Knuth, <i>The art of computer programming, Vol. III – Sorting and Searching</i> , ediția a doua, Addison Wesley Longman, 1998			
<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</li> </ul>	Examinare orală cu subiecte teoretice de sinteza și aplicative	40%

	<ul style="list-style-type: none"> <li>• 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>	Din oficiu	10%
10.5. Laborator	<ul style="list-style-type: none"> <li>• capacitatea de aplicare în practică, în contexte diferite, a cunoștințelor învățate;</li> <li>• capacitatea de a opera cu cunoștințele asimilate;</li> <li>• capacitatea de analiză, de interpretare personală;</li> </ul>	Verificari de laborator (probă practică: aplicații pe calculator)	15% 15%
		Participare activa la laborator si rezolvare teme de casa	20%
10.6. Proiect			
10.7. Standard minim de performanță			
<ul style="list-style-type: none"> <li>➤ Asimilarea limbajului de specialitate</li> <li>➤ Aplicarea adecvată în practică a structurilor algoritmice fundamentale</li> <li>➤ Implementarea și comentarea adecvată de unități de program în limbajul de programare C și folosirea eficientă a mediului de programare CodeBlocks</li> </ul>			

Data completării  
23 sept. 2024

Semnătura titularului de  
curs

Semnătura titularului de laborator

Data avizării în  
departament

24 sept. 2024

Director de departament  
Lector Dr. Anca Baci

Decan  
Prof. univ. dr. Mihaela Suditu