

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Inginerie Mecanică și Electrică
1.3. Departamentul	Automatică, Calculatoare și Electronică
1.4. Domeniul de studii universitare	Ingineria sistemelor
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Automatică și Informatică Aplicată

2. Date despre disciplină

2.1. Denumirea disciplinei	Proiectarea algoritmilor
2.2. Titularul activităților de curs	Șef lucrări dr. Mat. Cărbureanu Mădălina
2.3. Titularul activităților seminar/laborator	Șef lucrări dr. Mat. Cărbureanu Mădălina
2.4. Titularul activității proiect	-
2.5. Anul de studiu	1
2.6. Semestrul *	2
2.7. Tipul de evaluare	Examen
2.8. Categoria formativă** / regimul*** disciplinei	DS/DOB

* numărul semestrului este conform planului de învățământ;

** DF - Discipline fundamentale; DS - discipline de specializare; DC - discipline complementare

*** obligatorie/impusă = DOB; opțională = DOP; facultativă = DFA

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. Total ore studiu individual (studiu după suport de curs, bibliografie și notițe, documentare suplimentară în bibliotecă, pe platformele electronice de specialitate, pregătire seminar/laboratoare, teme, referate, portofolii și eseuri)							94
3.10. Total ore pe semestru							150
3.11. Numărul de credite							5

4. Condiții (acolo unde este cazul)

4.1. de curriculum	➤ Programarea calculatoarelor și limbaje de programare
4.2. de desfășurare a cursului	➤ Sală curs cu videoproiector, curs electronic și tipărit
4.3. de desfășurare a seminarului/laboratorului	➤ Laborator dotat cu tehnică de calcul și mediu de programare (CodeBlocks, Microsoft Visual Studio, Visual Studio Code, Online GDB beta, Online C++ Compiler, etc.) pentru limbajul C.

5. Competențe specifice acumulate și rezultatele învățării* care stau la baza acestora

Competențe profesionale	Rezultatele învățării*
Cp1. Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor.	A1- Studentul/absolventul utilizează argumentat conceptele din informatică și tehnologia calculatoarelor în rezolvarea de probleme bine definite din ingineria sistemelor și în aplicații ce impun utilizarea de hardware și software în sisteme industriale sau în sisteme informatice.

	<p>A2 - Studentul/absolventul utilizează limbaje, medii și tehnologii de programare și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) în rezolvarea de probleme bine definite din ingineria sistemelor</p> <p>RA1 - Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.</p>
<p>Cp5. Dezvoltarea de aplicații și implementarea algoritmilor și structurilor de conducere automată, utilizând principii de management de proiect, medii de programare și tehnologii bazate pe microcontrolere, procesoare de semnal, automate programabile, sisteme încorporate.</p>	<p>C1 - Studentul/absolventul descrie, identifică și sumarizează conceptele fundamentale și metodele de dezvoltare a limbajelor specifice dezvoltării de aplicații (secvențiale, concurente, timp real, non – timp real, locale, distribuite, încorporate, non – încorporate, mobile, online etc.) și de management de proiect.</p> <p>A3 - Studentul/absolventul evaluează modul de implementare a aplicațiilor de automatizare și informatică utilizând algoritmi și structuri de conducere automată, medii de programare și tehnologii bazate pe microcontrolere, procesoare de semnal, automate programabile, sisteme încorporate, inteligente etc.</p> <p>RA1 - Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.</p>

* C – cunoștințe; A – aptitudini; RA – responsabilitate și autonomie.

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

6.1. Obiectivul general al disciplinei	<p>➤ La sfârșitul cursului, studentul va fi capabil să dezvolte algoritmi proprii pe baza cunoștințelor asimilate (algoritmi fundamentali, metode de proiectare a algoritmilor, tipuri de date structurate).</p>
6.2. Obiectivele specifice	<p>La sfârșitul cursului, studentul va fi capabil:</p> <ul style="list-style-type: none"> ➤ să identifice și să definească conceptele (algoritm, program, alocare statică și dinamică de memorie, uniune, structură, stivă, coadă, arbori, grafuri, etc.) și metodele (recursivitate, Greedy, Divide et Impera, Backtracking, etc.) ce stau la baza proiectării algoritmilor; ➤ să explice conceptele și metodele ce stau la baza proiectării algoritmilor; ➤ să înțeleagă și să dezvolte aplicații utilizând algoritmi fundamentali de căutare și sortare, metodele de proiectare a algoritmilor, tipurile de date structurate (structuri, uniuni, enumerări, stive, cozi, grafuri, arbori); ➤ să dezvolte abilități de proiectare și implementare a algoritmilor (algoritmi de căutare, de sortare, de determinare a celor mai scurte drumuri, de determinare a arborelui parțial de cost minim, etc.) prezentați în cadrul cursului și a principalelor metode de proiectare a algoritmilor (recursivitate, Divide et Impera, Greedy, Backtracking);

	➤ să proiecteze și să implementeze algoritmi proprii pe baza cunoștințelor asimilate prin parcurgerea cursului.
--	---

7. Conținuturi

7.1. Curs	Nr. ore	Metode de predare	Observații
Algoritmi.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Recusivitate. Metodă principală de proiectare a algoritmilor.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Alocarea spațiului de memorie. Alocare statică și dinamică a memoriei. Pointeri. Operații cu pointeri.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Structuri, uniuni și enumerări. Definire noțiuni, declarare structuri, uniuni și enumerări. Exemple structuri, uniuni și enumerări. Accesul la câmpurile unei structuri, uniuni și enumerări. Diferența dintre structuri și uniuni. Declarații <i>typedef</i> .	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
<i>Stiva</i> -implementare dinamică. Definire noțiune <i>stivă</i> . Declarare <i>stivă</i> și tipuri de operații. Operații de bază (inițializare și creare <i>stivă</i> , parcurgerea elementelor stivei, ștergerea vârfului stivei, adăugarea unui element deasupra vârfului stivei) și alte tipuri de operații (adaugarea și ștergerea unui element în/din <i>stivă</i> după anumite criterii, etc.)	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
<i>Coadă</i> -implementare dinamică. Definire noțiune <i>coadă</i> . Declarare și tipuri de operații de bază (creare și parcurgere <i>coadă</i>) și alte tipuri de operații (adaugarea și ștergerea unui element în/din <i>coadă</i> după anumite criterii, etc.)	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)

Arbori. Arbori binari și de căutare. Definiere noțiuni specifice arbori. Modalități de reprezentare a arborilor în memoria calculatorului. Implementare dinamică arbori binari și de căutare. Operații de bază arbori binari și de căutare. Parcurgeri arbori.	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Grafuri. Grafuri neorientate și orientate. Paralelă noțiuni specifice grafuri neorientate și grafuri orientate. Modalități de reprezentare a grafurilor neorientate și orientate în memoria calculatorului. Algoritmul Roy-Warshall pentru determinarea matricei drumurilor asociată unui graf orientat. Algoritmi pentru determinarea celor mai scurte drumuri (Roy-Floyd, Dijkstra) într-un graf neorientat. Algoritmi pentru determinarea arborelui parțial de cost minim (Kruskal, Prim) pentru un graf neorientat.	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Algoritmi de căutare (secvențială, binară) și sortare (interschimbare, interclasare, selecție, numărare, inserție, sortare rapidă).	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Alte metode de proiectare a algoritmilor (Divide et Impera, Greedy, Backtracking).	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Bibliografie <ol style="list-style-type: none"> 1. Cărbureanu, M., <i>Elemente de proiectarea algoritmilor. Ghid teoretic și practic</i>. Editura Universității Petrol-Gaze din Ploiești, 2021. 2. Mueller J. P, Massaron, L., <i>Algorithms for Dummies</i>, Editia a II-a, John Wiley & Sons, New Jersey, 2022. 3. Helmut K., <i>Algorithms and Data Structures. Foundations and Probabilistic Methods for Design and Analysis</i>, Edition 1, Springer Cham, 2020. 4. Sedgewick, R., Wayne, K., <i>Algorithms</i>, Fourth Edition, Addison-Wesley, USA, 2011. 5. Skiena, S. S., <i>The algorithm design manual</i>, Second edition, Springer-Verlag, London, S.A., 2010. 6. Cormen, T, Leiserson, C., Rivest, R., Stein, C, <i>Introduction to Algorithm</i>, Third Edition, MIT Press, England, 2009. 7. Sedgewick, R., <i>Algorithms in C, Part 5: Graph Algorithms</i>, 3rd Edition, Addison-Wesley, Boston, San Francisco, 2002. 8. Knuth, D. E., <i>Arta programarii calculatoarelor: Algoritmi fundamentali</i>, Vol.1, Editura Teora, 1999. 			
7.2. Seminar / laborator	Nr. ore	Metode de predare	Observații

Dezvoltare pseudocod și scheme logice pentru diferite tipuri de aplicații (șirul lui Fibonacci, calcul $n!$, calcul $\text{cmmdc}(x, y)$, etc.)	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Implementare recursivă pentru diferite aplicații (suma cifrelor unui număr întreg n , calculul factorialului unui număr întreg n , determinarea celui de-al n -lea element din șirul lui Fibonacci, determinarea celui mai mare divizor comun (cmmdc) și a celui mai mic multiplu comun (cmmmc) dintre două numere întregi a și b , suma primelor n elemente ale unui vector de întregi, problema Turnurilor din Hanoi, etc.).	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații utilizând alocarea dinamică a memoriei (alocare dinamică vectori, alocare dinamică matrice, determinarea elementului maxim și respectiv minim dintr-un vector alocat dinamic, afișarea elementelor de pe diagonala principală și respectiv secundară a unei matrici alocate dinamic, etc.)	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații utilizând uniunile, structurile și enumerările (declarație structură <i>struct stud</i> și implementare set cerințe, declarație uniune <i>union angajat</i> și implementare set cerințe, declarație enumerare <i>enum discipline</i> și implementare set cerințe, etc.)	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații ce utilizează TDA <i>stivă</i> , implementare dinamică (concatenarea a două stive, deconcatenare stivă, construcția unei stive cu elemente prime dintr-o matrice, determinarea minimului și a maximului elementelor dintr-o listă simplu înlănțuită de tip stivă, inserarea între oricare două elemente ale unei stive a mediei aritmetice a elementelor învecinate, etc.)	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații ce utilizează TDA <i>coadă</i> , implementare	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator

dinamică (suma și produsul elementelor strict pozitive dintr-o coadă, construcție coadă cu elemente unei stive, construcția unui șir cu elemente prime din coadă, concatenarea a două cozi, etc.)			
Realizarea de diferite aplicații ce utilizează TDA arbore binar și de căutare, implementare dinamică (crearea și parcurgerea unui arbore binar, crearea și parcurgerea unui arbore binar de căutare, construcția unei cozi cu elementele unui arbore binar, construcția unui arbore binar cu elementele unui vector, determinarea maximului pe nivele într-un arbore echilibrat, etc.)	4	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații ce utilizează TDA graf- grafuri neorientate și orientate, implementare dinamică (verificarea dacă un graf neorientat reprezentat prin matricea de adiacență este complet, determinarea gradului interior și a gradului exterior al fiecărui nod dintr-un graf orientat, implementarea algoritmului Roy-Warshall pentru determinarea matricei drumurilor asociată unui graf orientat, implementarea algoritmului Roy-Floyd pentru determinarea celor mai scurte drumuri într-un graf neorientat, implementarea algoritmului Kruskall pentru determinarea arborelui parțial de cost minim asociat un graf neorientat, etc.)	4	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Implementarea algoritmilor clasici de căutare (secvențială și binară) și de sortare (Bubble Sort, Merge Sort, Selection Sort, Count Sort, Insertion Sort, etc.)	4	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Implementarea altor metode de proiectare a algoritmilor (implementare căutării binare prin metoda Divide et Impera, implementarea problemei restului de plată prin metoda Greedy,	4	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator

implementarea problemei generării permutărilor, aranjamentelor și combinărilor prin metoda Backtracking, etc.).			
Bibliografie <ol style="list-style-type: none"> 1. Cărbureanu, M., <i>Elemente de proiectarea algoritmilor. Ghid teoretic și practic</i>. Editura Universității Petrol-Gaze din Ploiești, 2021. 2. Vlădoiu. M., Constantinescu, Z., Moise, G., <i>Structuri de date fundamentale</i>, Editura Universității Petrol-Gaze din Ploiești, 2016. 3. Năchila, C., Dușmănescu, D., <i>Structuri de date și algoritmi în limbajul C/C++</i>, Solness, Timisoara, 2009. 4. Ciurea, E., Iolu, M., <i>Algoritmi. Teorie și aplicații</i>, Editura Universității Transilvania din Brașov, 2008. 5. Schildt, H., <i>C manual complet</i>, Editura Teora, 2002. 			
7.3. Proiect	Nr. ore	Metode de predare	Observații
Bibliografie -			

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

<ul style="list-style-type: none"> ➤ Disciplina este în concordanță cu planul de învățământ, studenții dobândind cunoștințe necesare la discipline (C++, etc.) care vor fi studiate în anii următori; ➤ Conținutul disciplinei este coroborat cu necesitățile angajatorilor și a asociațiilor profesionale ce activează în acest domeniu.

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Examen	Lucrare scrisă cu subiecte teoretice și aplicații	60%
		Frecvență curs	10%
9.5. Seminar/laborator	Activitate și verificare laborator	Examinare orală și program la calculator (rezolvarea unui subiect individual în limbajul C funcție de anumite cerințe)	30%

9.6. Proiect	-	-	-
9.7. Standard minim de performanță			
➤ Cunoașterea noțiunilor fundamentale și a metodelor de proiectare a algoritmilor, la nivel teoretic și practic în limbajul C.			

Data completării 24.09.2026	Semnătura titularului de curs	Semnătura titularului de seminar/laborator	Semnătura titularului de proiect
-----------------------------------	-------------------------------	---	----------------------------------

_____	_____	_____	_____
-------	-------	-------	-------

Data avizării în
departament

25.09.2026

Director de departament
Conf. dr. ing. Pricop Emil

Decan
Conf. dr. ing. Bădicioiu Marius
