

FIȘA DISCIPLINEI¹⁾

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	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Calculatoare

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	Verificare
2.8. Categoria formativă** / regimul*** disciplinei	DD/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 sinteză.

*** 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							50
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							15
Tutoriat							4
Examinări							10
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	➤ Programarea calculatoarelor și limbaje de programare
4.2. de competențe	➤ Operarea cu fundamente ale informaticii specifice programării calculatoarelor într-un limbaj de programare (limbajul C);

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 curs cu videoproiector, curs electronic și tipărit
5.2. de desfășurare a seminarului/laboratorului	➤ Laborator dotat cu tehnică de calcul și mediu de programare (Borland Turbo C, CodeBlocks, Microsoft Visual Studio, Online GDB beta, etc.) pentru limbajul C.

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none">➤ Cunoașterea și înțelegerea principalelor concepte și fundamente ale limbajului de programare structurată (limbajul C);➤ Operarea cu metode și concepte specifice programării structurate;➤ Cunoașterea și utilizarea limbajului de programare structurată pentru dezvoltarea de aplicații practice;➤ Proiectarea și implementarea propriilor algoritmi utilizând elemente ale programării structurate.
Competențe transversale	<ul style="list-style-type: none">➤ Comportarea onorabilă, responsabilă, în spiritul eticii profesionale și respectării legislației curente (inclusiv a drepturilor de proprietate intelectuală), pentru a asigura reputația statutului de student și a profesiei alese;➤ Demonstrarea spiritului de integrare, de inițiativă și de identificare a problemelor și responsabilităților din cadrul unei echipe de lucru interdisciplinare și plurispecializate;➤ Abilitatea de a conduce eficient un colectiv de lucru, de a comunica în bune condiții, de a lua decizii în timp real, de a distribui sarcini și verifica îndeplinirea acestora la toate nivelurile subordonate.

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

7.1. Obiectivul general al disciplinei	➤ 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).
7.2. 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.

8. Conținuturi

8.1. 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 memorie. Pointeri. Operații cu pointeri.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Structuri, uniuni și enumerări. Definiție noțiuni, declarare structuri, uniuni și enumerări. Acces câmpuri structuri, uniuni și enumerări. Diferența dintre structuri și uniuni.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Stiva-implementare dinamică. Declarare și tipuri de operații. Creare și parcurgere stivă. Ștergere și adăugare element din/în stivă.	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Coadă-implementare dinamică. Declarare și tipuri de operații. Creare și parcurgere coadă. Ștergere și adăugare element din/în coadă.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide-uri)
Arbori. Arbori binari și de căutare. Noțiuni specifice arbori. Modalități de reprezentare arbori în memorie calculator. 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. Noțiuni specifice grafuri. Modalități de reprezentare grafuri în memorie calculator. Parcurgeri ale grafurilor. Algoritmul Roy-Warshall. Algoritmi pentru determinarea celor mai scurte drumuri (Roy-Floyd, Dijkstra). Algoritmi pentru determinarea arborelui parțial de cost minim (Kruskal, Prim)	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).	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit

			și electronic (slide-uri)
<p>Bibliografie</p> <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. Knuth, D. E., <i>Tratat de programarea calculatoarelor. Algoritmi fundamentali</i>, Editura Tehnică, București, 1974. 3. Ciurea, E., Iolu, M., <i>Algoritmi. Teorie și aplicații</i>, Editura Universității Transilvania din Brașov, 2008. 4. Knuth, D. E., <i>The Art of Computer Programming: Fundamental algorithms</i>, Vol.1, Addison-Wesley Publisher, Massachusetts, 1997. 5. Knuth, D. E., <i>The Art of Computer Programming: Combinatorial Algorithms</i>, Volume 4A, Part 1, 1st Edition, Massachusetts, 2011. 6. Schildt, H., <i>C manual complet</i>, Editura Teora, 2002. 7. Jamsa, K., Klander, L., <i>Totul despre C și C++</i>. Manualul fundamental de programare în C și C++, Editura Teora, București, 1999. 8. Kernighan, B. W., Ritchie, D. M., <i>The C Programming Language</i>, Second Edition, Prentice Hall Software Series, 1988. 9. Sedgewick, R., Wayne, K., <i>Algorithms</i>, Fourth Edition, Addison-Wesley, USA, 2011. 10. Schildt, H., <i>C. The Complete Reference</i>, Fourth Edition, McGraw-Hill Companies, USA, 2000. 11. Wirth, N., <i>Algorithm +Data Structures=Programs</i>, Prentice Hall, USA, 1976. 12. Knuth, D. E., <i>Tratat de programarea calculatoarelor. Sortare și căutare</i>, Editura Tehnică, București, 1976. 			
8.2. 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 $n!$, determinarea și afișarea 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), factorialul unui număr întreg n , șirul lui Fibonacci, suma primelor n elemente ale unui vector de întregi, Turnurile din Hanoi, etc.).	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații utilizând alocarea dinamică de memorie (alocare dinamică vectori, alocare dinamică matrice, determinarea și afișarea elementelor prime dintr-un vector alocat dinamic, afișarea elementelor de pe diagonala principală și respectiv secundară a unei matrici alocată 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 struct stud și implementare set cerințe, declarație uniune union angajat și implementare set cerințe, declarație enumerare enum discipline 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 stivă, implementare dinamică (concatenarea a două stive, deconcatenare stivă, construcție stivă cu elemente prime din 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 coadă, implementare dinamică (construcție coadă cu elemente stivă, construcție șir cu elemente prime din coadă, concatenarea a două cozi, construirea unei liste simplu înlănțuite de tip coadă cu elementele un șir de întregi, eliminarea tuturor elementelor nule (de pe oricare poziție) dintr-o listă simplu înlănțuită de tip coadă, etc.)	2	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
Realizarea de diferite aplicații ce utilizează TDA arbore- arbore oarecare și de căutare, implementare dinamică (construcție coadă cu elemente arbore binar, construcție arbore binar cu elemente vector, maxim pe nivele într-un arbore echilibrat, crearea și parcurgerea în adâncime (RSD și SDR) a unui arbore oarecare, crearea și parcurgerea pe niveluri (BFS) a unui arbore binar de căutare)	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 (implementare Roy-Warshall, Roy-Floyd, implementare Kruskal, verificarea dacă un graf neorientat reprezentat prin matricea de adiacență este complet, parcurgerea	4	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator

în lățime (BFS) a unui graf neorientat, etc.)			
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 problema bacnotelor -metoda Greedy, problema Turnurilor din Hanoi- metoda Divide et Impera, generarea permutărilor – Backtracking, etc.).	4	clasică, centrată pe student și pe rezultatele învățării	Suport curs și laborator
<p>Bibliografie</p> <ol style="list-style-type: none"> Knuth, D. E., <i>Tratat de programarea calculatoarelor. Algoritmi fundamentali</i>, Editura Tehnică, București, 1974. Ciurea, E., Iolu, M., <i>Algoritmi. Teorie și aplicații</i>, Editura Universității Transilvania din Brașov, 2008. Cărbureanu, M., <i>Elemente de proiectarea algoritmilor. Ghid teoretic și practic</i>. Editura Universității Petrol-Gaze din Ploiești, 2021. Schildt, H., <i>C manual complet</i>, Editura Teora, 2002. Jamsa, K., Klander, L., <i>Totul despre C și C++</i>. <i>Manualul fundamental de programare în C și C++</i>, Editura Teora, București, 1999. Cormen, T, Leiserson, C., Rivest, R., Stein, C, <i>Introduction to Algorithm</i>, Third Edition, MIT Press, England, 2009. Roberts, E., <i>Thinking Recursively</i>, John Wiley&Sons, Inc., USA, 1986. Wirth, N., <i>Algorithms and Data Structures</i>, Prentice Hall, USA, 1985. Mueller J. P, Massaron, L., <i>Algorithms for Dummies</i>, John Wiley & Sons, New Jersey, 2017. Knuth, D. E., <i>Tratat de programarea calculatoarelor. Sortare și căutare</i>, Editura Tehnică, București, 1976. 			
8.3. Proiect	Nr. ore	Metode de predare	Observații
<p>Bibliografie</p> <p>-</p>			

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

- Disciplina este în concordanță cu planul de învățământ, studenții dobândind cunoștințe necesare la discipline (C++, Programare Web, etc.) care vor fi studiate în anii următori;
- Conținutul disciplinei este coroborat cu necesitățile angajatorilor și a asociațiilor profesionale ce activează în acest domeniu.

10.Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Verificare finală	Lucrare scrisă cu subiecte teoretice și aplicații	60%
		Frecvență curs	10%
10.5. Seminar/laborator	Activitate și verificare laborator	Examinare orală și program la calculator (rezolvarea unui subiect individual în limbajul C funcție de anumite cerințe)	30%
10.6. Proiect	-	-	-
10.7. Standard minim de performanță			
➤ Cunoașterea noțiunilor fundamentale și a metodelor de proiectare a algoritmilor, la nivel teoretic și practic în limbajul C.			

Data
completării
27.09.2021

Semnătura titularului de curs

Semnătura titularului de
seminar/laborator

Semnătura titularului de proiect

Data avizării în
departament

28.09.2021

Director de departament
Conf. dr. ing. PRICOP Emil
(Semnatură)

Decan
Conf. dr. ing. DINIȚĂ Alin
(Semnatură)