

# FIȘA DISCIPLINEI<sup>1)</sup>

## 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);

## 5. Condiții (acolo unde este cazul)

<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

5.1. 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) pentru limbajul C.

## 6. Competențe specifice acumulate

<b>Competențe profesionale</b>	<ul style="list-style-type: none"> <li>➤ Cunoașterea și înțelegerea principalelor concepte și fundamente ale limbajului de programare structurată (limbajul C);</li> <li>➤ Operarea cu metode și concepte specifice programării structurate;</li> <li>➤ Cunoașterea și utilizarea limbajului de programare stucturată pentru dezvoltarea de aplicații practice;</li> <li>➤ Proiectarea și implementarea propriilor algoritmi utilizând elemente ale programării structurate.</li> </ul>
<b>Competențe transversale</b>	<ul style="list-style-type: none"> <li>➤ 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;</li> <li>➤ Demonstrarea spiritului de integrare, de inițiativă și de identificare a problemelor și responsabilităților din cadrul unei echipe de lucru interdisciplinare și plurispecializate;</li> <li>➤ Abilitatea de a comunica în bune condiții și de a lua decizii în timp real.</li> </ul>

## 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"> <li>➤ să identifice și să definească conceptele (algoritm, program, alocare statică și dinamică de memorie, etc.) și metodele (recursivitate, metoda Greedy, Divide et Impera, Backtracking, etc.) ce stau la baza proiectării algoritmilor;</li> <li>➤ să explice conceptele și metodele ce stau la baza proiectării algoritmilor;</li> <li>➤ să înțeleagă și să dezvolte aplicații utilizând algoritmii fundamentali, metodele de proiectare a algoritmilor, tipurile de date structurate (structuri, uniuni, enumerări, liste alocate dinamic (stive, cozi), grafuri, arbori);</li> <li>➤ să dezvolte abilități de proiectare și implementare a algoritmilor (algoritmi de căutare, de sortare, de determinare a arborelui parțial de cost minim, etc.)</li> </ul>

	<p>prezențați în cadrul cursului și a principalelor metode de proiectare a algoritmilor;</p> <p>➤ să proiecteze și să implementeze algoritmi proprii pe baza cunoștințelor asimilate prin parcurgerea cursului.</p>
--	---

## 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)
Recusivitate. Metodă principală de proiectare a algoritmilor.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Alocarea memoriei. Operații cu pointeri, tipuri de date structurate. Structuri, declarații typedef.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Uniuni, enumerări. Șiruri de caractere. Stiva-tip particular de listă.	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Coada-tip particular de listă.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Arbori. Arbori binari.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Grafuri. Grafuri neorientate.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Grafuri. Grafuri orientate.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Algoritmi clasici de căutare.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)

Algoritmi fundamentali de sortare.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Algoritmi de determinare a arborelui parțial de cost minim și a celor mai scurte drumuri într-un graf neorientat.	2	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)
Metode de proiectare a algoritmilor (metoda Greedy, Divide et Impera, Backtracking).	4	Interactivă și convențională, centrată pe student	Suport curs în format tipărit și electronic (slide)

#### Bibliografie

1. Knuth, D.E., *Arta programarii calculatoarelor*, Vol. 1, Algoritmi fundamentali, Editura Teora, 2000;
2. Knuth, D. E., *Arta programarii calculatoarelor*, Vol. 3, Sortare si cautare, Editura Teora, 2002;
3. Schildt, D., *C manual complet*, Editura Teora, 2000;
4. Cormen, T., *Introducere in algoritmi*, Editura Byblos, 2004;
5. Cristea, V., Athanasiu, I., Kalisz, E., Iorga, V., *Tehnici de programare*, Editura Teora, Bucuresti, 1993;
6. Georgescu, H., *Tehnici de programare*, Editura Universitatii din Bucuresti, 2005;
7. Tomescu, I., *Data structures*, Editura Universitatii din Bucuresti, 1997.

<b>8.2. Seminar / laborator</b>	Nr. ore	Metode de predare	Observații
Recapitulare limbaj C (vectori, matrici).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Recusivitatea (factorialul unui număr întreg n, șirul lui Fibonacci, cel mai mic multiplu comun dintre două numere întregi a și b, suma primelor n elemente ale unui vector de întregi, Turnurile din Hanoi, etc.).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Structuri, declarații typedef și șiruri de caractere (probleme ce utilizează diferite tipuri de structuri).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Stiva- tip particular de listă (operații de bază, diferite tipuri de probleme).	4	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Cooda-tip particular de listă (operații de bază, diferite tipuri de probleme).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs

Arbori binari (operații de bază, diferite tipuri de probleme).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Grafuri neorientate (implementare metode de reprezentare în memorie, diferite tipuri de probleme, algoritmi de determinare a arborelui parțial de cost minim și a celor mai scurte drumuri).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Grafuri orientate (implementare metode de reprezentare în memorie, diferite tipuri de probleme).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Algoritmi clasici de cautare (implementare căutare secvențială, căutare binară).	2	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Algoritmi fundamentali de sortare (implementare sortare prin interschimbare, interclasare, selecție, numărare și inserție).	4	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
Metode de proiectare a algoritmilor (implementare problema bacnotelor -metoda Greedy, problema Turnurilor din Hanoi- metoda Divide et Impera, generarea permutărilor – Backtracking, etc.).	4	Interactivă, centrată pe student și pe rezultatele învățării	Suport curs
<p>Bibliografie</p> <ol style="list-style-type: none"> <li>1. Knuth, D.E., <i>Arta programarii calculatoarelor</i>, Vol. 1, Algoritmi fundamentali, Editura Teora, 2000;</li> <li>2. Knuth, D. E., <i>Arta programarii calculatoarelor</i>, Vol. 3, Sortare si cautare, Editura Teora, 2002;</li> <li>3. Schildt, D., <i>C manual complet</i>, Editura Teora, 2000.</li> </ol>			
<b>8.3. Proiect</b>	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) 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	Cunoașterea tipurilor de date structurate, a principalelor metode de proiectare a algoritmilor și a algoritmilor fundamentali- <b>Evaluare parțială</b>	Lucrare scrisă sub formă de text (10 întrebări cu răspuns deschis și două aplicații software) cu subiecte teoretice și aplicații în limbajul C;	<b>60%</b>
		<b>Frecvență curs</b>	<b>10%</b>
10.5. Seminar/laborator	<b>Verificare laborator</b>	Rezolvarea unui subiect individual (a unei aplicații în limbajul C), funcție de anumite cerințe;	<b>30%</b>
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  
10.09.2020

Semnătura titularului de curs

Semnătura titularului de  
seminar/laborator

Semnătura titularului de proiect

Data avizării în  
departament

28.09.2020

Director de departament  
(funcție didactică, nume, prenume)  
(Semnatură)

\_\_\_\_\_  
Decan  
(funcție didactică, nume, prenume)  
(Semnatură)