

# FIȘA DISCIPLINEI<sup>1</sup>

## 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea <sup>2</sup> / Departamentul <sup>3</sup>	Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod <sup>4</sup> )	INGINERIE ȘI MANAGEMENT / 230
1.5 Ciclul de studii	Licență
1.6 Programul de studii (denumire/cod/calificarea)	INGINERIE ECONOMICĂ ÎN INDUSTRIA CHIMICĂ ȘI DE MATERIALE / 70 / Inginer

## 2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă <sup>5</sup>	Programarea calculatoarelor și limbaje de programare / DF						
2.2 Titularul activităților de curs	Șef lucr. dr. ing. Berdie Adela Diana						
2.3 Titularul activităților aplicative <sup>6</sup>	Asist.dr.ing.Raț Cezara						
2.4 Anul de studii <sup>7</sup>	I	2.5 Semestrul	I	2.6 Tipul de evaluare	D	2.7 Regimul disciplinei <sup>8</sup>	DI

## 3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)<sup>9</sup>

3.1 Număr de ore asistate integral/săptămână	4 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	2
3.1* Număr total de ore asistate integral/sem.	56 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	28
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	4,93 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1,5
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			2,43
3.7* Număr total de ore activități neasistate/semestru	69 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			21
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			34
3.8 Total ore/săptămână <sup>10</sup>	8,93				
3.8* Total ore/semestru	125				
3.9 Număr de credite	5				

## 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Cunoștințe de matematică elementară – nivel liceu
4.2 de competențe	•

<sup>1</sup> Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 și cerințelor Standardelor specifice ARACIS valabile începând cu 01.10.2017.

<sup>2</sup> Se înscrie numele facultății care gestionează programul de studii căruia îi aparține disciplina.

<sup>3</sup> Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

<sup>4</sup> Se înscrie codul prevăzut în HG nr.140/16.03.2017 sau în HG similare actualizate anual.

<sup>5</sup> Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină fundamentală (DF), disciplină de domeniu (DD), disciplină de specialitate (DS) sau disciplina complementară (DC).

<sup>6</sup> Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

<sup>7</sup> Anul de studii în care este prevăzută disciplina în planul de învățământ.

<sup>8</sup> Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI), disciplină opțională (DO) sau disciplină facultativă (Df).

<sup>9</sup> Numărul de ore de la rubricile 3.1\*, 3.2\*,...,3.8\* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,..., 3.8. Informațiile din rubricile 3.1, 3.4 și 3.7 sunt chei de verificare folosite de ARACIS sub forma: (3.1)+(3.4) ≥ 28 ore/săpt. și (3.8) ≤ 40 ore/săpt.

<sup>10</sup> Numărul total de ore / săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.7.

## 5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> <li>• Sală de curs echipată cu videoproiector și conexiune la Internet.</li> <li>• Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise.</li> <li>• Nu se acceptă părăsirea sălii de curs fără aprobarea cadrului didactic</li> <li>• Predarea în format online a cursului se desfășoară folosind uneltele CVUPT prin intermediul Campusului Virtual (CV) și al aplicației Zoom (videoconferințe)</li> <li>• Toate resursele educaționale utilizate pentru desfășurarea cursurilor sunt încărcate pe platforma CV</li> </ul>
5.2 de desfășurare a activităților practice	<ul style="list-style-type: none"> <li>• Sală de laborator echipată cu computere</li> <li>• Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise.</li> <li>• Nu se acceptă părăsirea sălii de desfășurare a activității practice fără aprobarea cadrului didactic</li> <li>• Desfășurarea în format online a activităților practice (laborator) se realizează folosind uneltele CVUPT prin intermediul Campusului Virtual (CV) și al aplicației Zoom (videoconferințe)</li> <li>• Toate resursele educaționale utilizate pentru desfășurarea laboratoarelor sunt încărcate pe platforma CV</li> </ul>

## 6. Competențe la formarea cărora contribuie disciplina

Competențe specifice	<ul style="list-style-type: none"> <li>• Operarea cu fundamente matematice, ingineresti și ale informaticii</li> <li>• Selectarea de software dedicat pentru editarea documentelor și proiectelor de execuție ingineresti precum și dobândirea cunoștințelor privind limbajele de programare și operarea cu mediile de programare specifice</li> </ul>
Competențele profesionale în care se înscriu competențele specifice	<p>Cunoștințe teoretice ce trebuie cunoscute după parcurgerea disciplinei:</p> <ul style="list-style-type: none"> <li>• să descrie algoritmi de rezolvare pentru probleme de calcul uzuale în domeniul ingineriei.</li> <li>• să înțeleagă modul de rezolvare a problemelor dezvoltând aplicații în limbajul (C)</li> <li>• Să cunoască sintaxa limbajului C și posibilitățile oferite de acest limbaj</li> </ul> <p>Deprinderi dobândite după parcurgerea disciplinei:</p> <ul style="list-style-type: none"> <li>• să descrie un algoritm prin schemă logică și limbaj pseudocod</li> <li>• să dezvolte aplicații în limbajul C</li> </ul> <p>Abilități ce trebuie dobândite după parcurgerea disciplinei:</p> <ul style="list-style-type: none"> <li>• să utilizeze mediile de programare Borland C și Code Bloks</li> </ul>
Competențele transversale în care se înscriu competențele specifice	<ul style="list-style-type: none"> <li>• CT3</li> </ul> <p>Identificarea oportunităților de formare continuă și utilizarea eficientă, pentru propria dezvoltare, a surselor informaționale și a resurselor de comunicare și formare profesională asistată (portaluri Internet, aplicații software de specialitate, baze de date, cursuri on-line etc.) atât în limba română, cât și într-o limbă de circulație internațională</p>

## 7. Obiectivele disciplinei (asociate competențelor de la punctul 6)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Dezvoltarea capacităților studenților de a dezvolta o gândire algoritmică în abordarea rezolvării problemelor ingineresti</li> <li>• Familiarizarea cu un limbaj de programare procedural – limbajul C – și cu paradigmele programării procedurale prin realizarea de aplicații specifice</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Dobândirea noțiunilor specifice gândirii algoritmice în scopul soluționării problemelor aplicative</li> <li>• Însușirea principiilor ce stau la baza programării</li> <li>• Însușirea unei gândiri algoritmice și dezvoltarea abilităților de proiectare de soluții algoritmice</li> <li>• Inițiere în utilizarea limbajului de programare C prin implementarea algoritmilor dezvoltați</li> </ul>

## 8. Conținuturi<sup>11</sup>

8.1 Curs	Număr de ore	Metode de predare <sup>12</sup>
1. Algoritmi 1.1 Noțiuni generale 1.2 Etapele rezolvării unei probleme – enunț, date intrare/ ieșire 1.3 Obiectele cu care lucrează algoritmi și operații permise 1.4 Operații pe care le efectuează un algoritm	2	Prelegere susținută prin prezentări PPT, conversații, explicații, exemplificări
2. Principiile programării structurate 2.1 Generalități 2.2 Moduri de reprezentare a algoritmilor 2.3 Structuri de bază și descrierea lor	4	
3. Elemente de bază ale limbajului C 3.1 Noțiuni introductive 3.2 Prezentarea mediului de programare Code Blocks. Primul program 3.3 Structura programelor – de la cod sursă la program executabil 3.4 Funcții standard de intrare/ieșire	2	
4. Vocabularul limbajului C 4.1 Noțiuni de bază 4.2 Nume de variabile 4.3 Tipuri și dimensiuni de date 4.4 Constante, declarații 4.5 Expresii 4.6 Operatori	2	
5. Controlul fluxului de execuție. Instrucțiuni 5.1 Instrucțiunea expresie 5.2 Instrucțiunea IF 5.3 Instrucțiunea compusă 5.4 Instrucțiunea SWITCH 5.5 Instrucțiunile FOR, WHILE și DO WHILE 5.6 Instrucțiuni de salt – BREAK, CONTINUE, GOTO, RETURN	8	
6. Tablouri. 6.1 Tablouri unidimensionale 6.2 Tablouri bidimensionale 6.3 Șiruri de caractere	6	
7. Tipul de date structura	2	
8. Recapitulare	2	

Bibliografie<sup>13</sup>

1. A.D. Berdie – Material electronic curs – Campus Virtual UPT, 2023
2. Anca Iordan, Daniela Cristea, Adela Berdie, Programarea calculatoarelor, Editura Mirton, 2004
3. Brian W. Kernighan, Denis Ritchie, Limbajul C, Ed. Teora, 2003
4. Dr. Kris Jamsa & Lars Klander, Totul despre C și C++. Manualul fundamental de programare în C și C++, Ed. Teora, 2001
5. Liviu Negrescu, Limbajele C și C++ pentru începători, vol. I, Limbajul C, Ed. MicroInformatica, 1994
6. V.Iorga, Programare în C, Ed.Albastră, Cluj Napoca, 2011

<sup>11</sup> Se detaliază toate activitățile didactice prevăzute prin planul de învățământ (tematicile prelegerilor și ale seminariilor, lista lucrărilor de laborator, conținuturile etapelor de elaborare a proiectelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

<sup>12</sup> Prezentarea metodelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

<sup>13</sup> Cel puțin un un titlu trebuie să aparțină colectivului disciplinei iar cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, de circulație națională și internațională, existentă în biblioteca UPT.

8.2 Activități aplicative <sup>14</sup>	Număr de ore	Metode de predare
1. Dezvoltarea corectă a algoritmilor din specificații. Evaluarea expresiilor	2	Expunere temă, discuții, întrebări, îndrumare în rezolvarea pe calculator a problemelor
2. Reprezentarea algoritmilor prin limbaj pseudocod	3	
3. Reprezentarea algoritmilor prin scheme logice	3	
4. Editarea și testarea aplicațiilor în limbaj C cu ajutorul utilitarului Code Blocks. Programe simple cu funcții de intrare/ieșire standard	2	
5. Utilizarea instrucțiunilor IF și SWITCH	4	Expunere temă, discuții, întrebări, îndrumare în rezolvarea pe calculator a problemelor
6. Utilizarea instrucțiunii FOR, WHILE și DO WHILE	6	Expunere temă, discuții, întrebări, îndrumare în rezolvarea pe calculator a problemelor
7. Prelucrări asupra tablourilor unidimensionale, bidimensionale și a șirurilor de caractere	4	Expunere temă, discuții, întrebări, îndrumare în rezolvarea pe calculator a problemelor
8. Definirea și utilizarea structurilor pentru memorarea datelor	2	Expunere temă, discuții, întrebări, îndrumare în rezolvarea pe calculator a problemelor
9. Recapitulare și recuperare	2	Expunere teme, discuții, întrebări, îndrumare în rezolvarea pe calculator a problemelor
Bibliografie <sup>15</sup> 1. A.D.Berdie, Material electronic laborator – Campus Virtual UPT, 2023 2. A. Iordan, D. Cristea, A. Berdie, Programarea calculatoarelor, Editura Mirton, Timișoara, 2004 3. I. Muscalagiu, D. Muscalagiu, M. Pănoiu, Culegere de probleme în limbajul C, Editura Mirton, Timișoara, 2001 4. H. Ciocârlie, R. Ciocârlie, Tehnici de programare și structuri de date, Editura Eurostampa, 2012 5. V. Iorga, Programare în C, Editura Albastră, 2011		

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

<ul style="list-style-type: none"> <li>• Cunoștințele de algoritmică și programare sunt importante pentru toate materiile cu specific software care fac parte din planul de învățământ al specializării</li> <li>• Majoritatea angajatorilor reprezentativi din domeniul aferent programului solicită atât cunoștințe programare și cunoașterea unui limbaj de programare</li> <li>•</li> </ul>
---

**10. Evaluare**

<sup>14</sup> Tipurile de activități aplicative sunt cele precizate în nota de subsol 5. Dacă disciplina conține mai multe tipuri de activități aplicative atunci ele se trec consecutiv în liniile tabelului de mai jos. Tipul activității se va înscrie într-o linie distinctă sub forma: „Seminar:”, „Laborator:”, „Proiect:” și/sau „Practică:”.

<sup>15</sup> Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

Tip activitate	10.1 Criterii de evaluare <sup>16</sup>	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	Scris - subiecte teoretice și aplicații (două părți de evaluare – algoritmi și limbaj C / Test online prin intermediul platformei CV – două părți de evaluare –subiecte teoretice și aplicații cu probleme de algoritmi și limbaj C)	0,66
10.5 Activități aplicative	<b>S:</b>		
	<b>L:</b> Abilități în rezolvarea aplicațiilor de laborator	Oral -verificare la încheierea activității de laborator / Test online prin intermediul platformei CV – aplicații cu probleme din algoritmi si limbajul C	0,34
	<b>P<sup>17</sup>:</b>		
	<b>Pr:</b>		
<b>10.6 Standard minim de performanță</b> (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor <sup>18</sup> )			
<ul style="list-style-type: none"> <li>• Proiectarea algoritmului, testarea și executarea unui program de complexitate medie</li> <li>• Operații de bază în limbajul C</li> <li>• Programele de pe biletul de examen trebuie să fie corecte și să rezolve minimul de cerințe solicitat</li> <li>• Testele online de la laborator și evaluarea distribuită, susținute prin intermediul platformei CV, conțin subiecte teoretice și aplicații de tip probleme cu complexitate medie</li> </ul>			

**Data completării**

05.10.2023

**Director de departament  
(semnătura)**

.....



**Titular de curs  
(semnătura)**

.....

**Data avizării în Consiliul Facultății<sup>19</sup>**

16.10.2023

**Titular activități aplicative  
(semnătura)**

.....

**Decan  
(semnătura)**



<sup>16</sup> Fișele disciplinelor trebuie să conțină procedura de evaluare a disciplinei cu precizarea criteriilor, a metodelor și a formelor de evaluare, precum și cu precizarea ponderilor atribuite acestora în nota finală. Criteriile de evaluare se formulează în mod distinct pentru fiecare activitate prevăzută în planul de învățământ (curs, seminar, laborator, proiect). Ele se vor referi și la formele de verificare pe parcurs (teme de casă, referate ș.a.)

<sup>17</sup> În cazul când proiectul nu este o disciplină distinctă, în această rubrică se va preciza și modul în care rezultatul evaluării proiectului condiționează admiterea studentului la evaluarea finală din cadrul disciplinei.

<sup>18</sup> Nu se va explica cum se acorda nota de promovare.

<sup>19</sup> Avizarea este precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii cu privire la fișa disciplinei.