

## FIȘA DISCIPLINEI

### *Programare imperativă/ procedurală*

#### 1. Date despre program

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918”
1.2. Facultatea	Științe Exacte și Inginerești
1.3. Departamentul	Științe Exacte și Inginerești
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Licență
1.6. Programul de studii	Informatica

#### 2. Date despre disciplină

2.1. Denumirea disciplinei	<i>Programare imperativă/ procedurală</i>	2.2. Cod disciplină	INFO 103
2.3. Titularul activității de curs	Domșa Ovidiu		
2.4. Titularul activității de seminar	Incze Arpad		
2.5. Anul de studiu	<b>I</b>	2.6. Semestrul	<b>I</b>
		2.7. Tipul de evaluare (E/C/VP)	<b>E</b>
		2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)	<b>O</b>

#### 3. Timpul total estimat

3.1. Numar ore pe saptamana	<b>4</b>	din care: 3.2. curs	<b>2</b>	3.3. seminar/laborator	<b>3</b>
3.4. Total ore din planul de învățământ	<b>70</b>	din care: 3.5. curs	<b>28</b>	3.6. seminar/laborator	<b>42</b>
Distribuția fondului de timp					Ore
Studiul după manual, suport de curs, bibliografie și notițe					<b>50</b>
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					<b>50</b>
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					<b>50</b>
Tutoriat					-
Examinări					<b>28</b>
Alte activități .....					-

3.7 Total ore studiu individual	<b>108</b>
3.8 Total ore din planul de învățământ	<b>70</b>
3.9 Total ore pe semestru	<b>178</b>
3.10 Numărul de credite	<b>6</b>

#### 4. Precondiții (acolo unde este cazul)

4.1. de curriculum	
4.2. de competențe	

#### 5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	<i>Sala dotata cu videoproiector/tabla</i>
5.2. de desfășurarea a seminarului/laboratorului	<i>Laboratoare – calculatoare dotate cu: BorlandC, Internet.</i>

## 6. Competențe profesionale si transversale

Competențe explicitate prin descriptorii de nivel (C1)	<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 soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază</li> <li>- C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.</li> <li>- C 1.4 Testarea unor aplicații pe baza unor planuri de test.</li> <li>- C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente.</li> </ul>
Competențe specifice	<ul style="list-style-type: none"> <li>- Formarea de aptitudini necesare în rezolvarea problemelor elementare folosind algoritmi;</li> <li>- Identificarea problemelor și elaborarea metodelor algoritmice de rezolvare;</li> <li>- Studentul absolvent al acestui curs va fi capabil să transpună în limbaj algoritmic (pseudocod, limbaj de programare) rezolvarea algoritmică a unei probleme.</li> <li>- Aprofundarea conceptelor de programare structurată și programare procedurală.</li> </ul>
Competențe transversale	<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>CT2 Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatice de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse</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> <p><b>Competențe cognitive:</b> dobândirea de cunoștințe fundamentale și specifice privind conceptele fundamentale de algoritm, capacitatea de identificare a aplicabilității algoritmilor studiați în probleme reale; înțelegerea necesității utilizării algoritmilor în abordarea problemelor: dobândirea de cunoștințe fundamentale privind modalități de reprezentare și transcriere a algoritmilor;</p> <p><b>Competențe afectiv valorice:</b> dezvoltarea capacității de analiză și înțelegere a unei probleme din perspectivă algoritmică în mod eficient. Spirit de echipă (încurajarea studentului de a lucra în echipe de proiectare, analiză și programare); conștientizarea importanței cunoașterii și aprofundării algoritmilor elementari.</p>

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- <i>Dezvoltarea gândirii algoritmice și a abilităților de a elabora algoritmi de rezolvare a problemelor pornind de la cerințele și datele de intrare ale acesteia.</i></li> <li>- <i>Însușirea instrumentelor de bază în elaborarea algoritmilor.</i></li> </ul>
---------------------------------------	---

	<ul style="list-style-type: none"> <li>- Cunoașterea algoritmilor elementari și a metodelor elementare de elaborare a algoritmilor .</li> <li>- Deprinderea studenților cu utilizarea unui limbaj de programare evoluat în vederea implementării algoritmilor elaborați.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Cunoașterea principiilor, a conceptelor elementare ale gândirii algoritmice precum și a modalităților de reprezentare ale algoritmilor.</li> <li>• Cunoașterea noțiunilor elementare privind limbajul pseudocod și scheme logice.</li> <li>• Cunoașterea principiilor programării structurate, programării procedurale.</li> <li>• Elaborarea de modele matematice și transcrierea lor în algoritmi pentru tipurile clasice de probleme.</li> <li>• Însușirea principalelor modalități de reprezentare a datelor și modul de organizare a acestora. Principii generale și modalități concrete de reprezentare în C. Cunoașterea mediului de programare C.</li> <li>• Realizarea de aplicații, rezolvarea de probleme folosind instrucțiunile limbajului de programare.</li> </ul>

## 8. Conținuturi

Curs	Metode de predare	Observații
I. Principii ale programării imperative procedurale. 1. Elemente pentru descrierea algoritmilor și etape în elaborarea programelor (Scurt istoric, Noțiunea de algoritm. Caracteristici. Descrierea algoritmilor. Proiectarea algoritmilor.)	<i>Prelegere, discuții, exemplificări</i>	
2. Principiile de programare. Programarea structurată. Programarea modulară. Programarea orientată-obiect.	<i>Prelegere, discuții, exemplificări</i>	
3. Moduri de organizare și reprezentare a datelor	<i>Prelegere, discuții, exemplificări</i>	
4. Metode de descriere a algoritmilor. Scheme logice. Limbaj pseudocod. Limbaje de programare.	<i>Prelegere, discuții, exemplificări</i>	
5. Elaborarea, corectitudinea, complexitatea și testarea algoritmilor	<i>Prelegere, discuții, exemplificări</i>	
6. Algoritmi elementari. Schimbarea valorilor a două variabile. Parcurgerea elementelor unei mulțimi. Implementarea cuantificatorilor matematici oricare și există. Implementarea produsului cartezian a două mulțimi.	<i>Prelegere, discuții, exemplificări</i>	
Evaluare parțială	<i>Test de evaluare, Grila</i>	
7. Metode de sortare. Sortarea prin metoda bulelor, algoritmul “Bubble Sort”. Sortarea prin selecție. Sortarea prin inserție. Sortarea prin numărare	<i>Prelegere, discuții, exemplificări</i>	
8. Metode elementare de elaborare a algoritmilor.	<i>Prelegere, discuții, exemplificări</i>	

Interclasarea.		
9. Recursivitatea. Funcții recursive	<i>Prelegere, discuții, exemplificări</i>	
II. Elemente de bază ale limbajului C 1. Vocabularul limbajului. Tipuri de date 2. Programarea structurata. Instrucțiuni C. Secvențe liniare. Instrucțiuni alternative. Instrucțiuni repetitive	<i>Prelegere, discuții, exemplificări</i>	
3. Funcții definite de utilizator. Biblioteci de funcții. Domeniul de vizibilitate al variabilelor, Apelul recursiv.	<i>Prelegere, discuții, exemplificări</i>	
III. Algoritmi elementari. Aplicații	<i>Prelegere, discuții, exemplificări</i>	
1. Aplicații în limbajul C. Comentate, prezentate, discuții pe tema soluțiilor	<i>Prelegere, discuții, exemplificări</i>	
<b>Bibliografie</b>		
<ol style="list-style-type: none"> <li>Andone R., Gârbacea I., Algoritmi fundamentali o perspectivă C++, Editura Libris, Cluj Napoca, 1995.</li> <li>Cormen T.H., Leiserson E.C., Rivest R.R., Introducere în algoritmi, Editura Libris Agora, 2000 (traducere în limba română).</li> <li>Dahl O.J., Dijkstra E.W., Hoare C.A.R., Structured Programing, Academic Press, 1972.</li> <li>Domșa O., Bazele algoritmilor, Curs, Seria Didactică, Univ. 1 Decembrie 1918 Alba Iulia, 2011.</li> <li>Rotar C., Algoritmi și structuri de date, Curs, Seria Didactică, Univ. 1 Decembrie 1918 Alba Iulia, 2003.</li> <li>Logofatu D., Algoritmi fundamentali în C++, Ed. Polirom, 2007.</li> <li>Donald E. Knuth, <a href="#">The Art of Computer Programming</a>, Volumes 1–3, Addison-Wesley Professional Volume 1: Fundamental Algorithms (3rd edition), 1997. Addison-Wesley Professional, Volume 2: Seminumerical Algorithms (3rd Edition), 1997. Addison-Wesley Professional, Volume 3: Sorting and Searching (2nd Edition), 1998. Addison-Wesley Professional.</li> </ol>		
<b>Seminar-laborator</b>		
1. Scurt istoric, Noțiunea de algoritm. Caracteristici. Descrierea algoritmilor. Proiectarea algoritmilor.	<b>Lucrare practică de laborator</b>	
2. Scheme logice. Limbaj pseudocod. Limbaje de programare. Structuri de control: liniară, alternativă, repetitivă.	<b>Lucrare practică de laborator</b>	
3. Elemente specifice ale limbajului C. Medii de dezvoltare a programelor. BorlandC, Code Blocks for C	<b>Lucrare practică de laborator</b>	
4. Structura programelor C. Reprezentarea datelor în limbaj C. Citirea/Scrierea datelor în C. Conversii de date în C.	<b>Lucrare practică de laborator</b>	
5. Instrucțiuni C. Instrucțiunea IF, CASE, FOR, WHILE	<b>Lucrare practică de laborator</b>	
6. Structuri de date simple și structurate (tablouri)	<b>Lucrare practică de laborator</b>	
7. Structuri de date simple și structurate (string)	<b>Lucrare practică de laborator</b>	
8. Schimbarea valorilor a două variabile. Parcurgerea elementelor unei mulțimi. Implementarea cuantificatorilor matematici oricare și există. Implementarea produsului cartezian a două mulțimi.	<b>Lucrare practică de laborator</b>	
9. Sortarea prin metoda bulelor, algoritmul “Bubble Sort”. Sortarea prin selecție. Sortarea prin inserție. Sortarea prin numărare.	<b>Lucrare practică de laborator</b>	
10. Interclasare. Probleme practice care implementează algoritmi studiați	<b>Lucrare practică de laborator</b>	

11. Verificarea programelor in C	<b>Lucrare practică de laborator</b>	
12. Recursivitatea. Funcții recursive.	<b>Lucrare practică de laborator</b>	
13. Aplicații și probleme recapitulative	<b>Lucrare practică de laborator</b>	
14. Evaluarea portofoliului practic de laborator	<b>Evaluare practică</b>	

### **Bibliografie**

1. Andone R., Gârbacea I., Algoritmi fundamentali o perspectivă C++, Editura Libris, Cluj Napoca, 1995.
2. Cormen T.H., Leiserson E.C., Rivest R.R., Introducere în algoritmi, Editura Libris Agora, 2000 (traducere în limba română).
3. Dahl O.J., Dijkstra E.W., Hoare C.A.R., Structured Programming, Academic Press, 1972.
4. Domșa O., Bazele algoritmilor, Curs, Seria Didactică, Univ. 1 Decembrie 1918 Alba Iulia, 2011.
5. Rotar C., Algoritmi și structuri de date, Curs, Seria Didactică, Univ. 1 Decembrie 1918 Alba Iulia, 2003.
6. Logofatu D., Algoritmi fundamentali in C++, Ed. Polirom, 2007.
7. Donald E. Knuth, The Art of Computer Programming, Volumes 1–3, Addison-Wesley Professional Volume 1: Fundamental Algorithms (3rd edition), 1997. Addison-Wesley Professional, Volume 2: Seminumerical Algorithms (3rd Edition), 1997. Addison-Wesley Professional, Volume 3: Sorting and Searching (2nd Edition), 1998. Addison-Wesley Professional.

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

- Nu e cazul.

### **10. Evaluare**

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<i>Evaluare finala</i>	<i>Examen scris/ practic</i>	50%
	-	-	-
10.5 Seminar/laborator	<i>Verificare pe parcurs</i>	<i>Portofoliu de lucrări practice de laborator</i>	50%
	-	-	-
10.6 Standard minim de performanță:			
Minim 25%+25%= 50 %			

Data completării

.....

Semnătura titularului de curs

.....

Semnătura titularului de seminar

.....

Data avizării în catedră

.....

Semnătura director de departament

.....