

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea ² / Departamentul ³	Automatică și Calculatoare/ Calculatoare și Tehnologia Informației
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod ⁴)	Calculatoare și Tehnologia Informației / 10
1.5 Ciclu de studii	Licență
1.6 Programul de studii (denumire/cod/calificarea)	Tehnologia Informației / 20 / Inginer

2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă ⁵	Tehnici de Programare / DF						
2.2 Titularul activităților de curs	Prof. dr. ing. Horia Ciocârlie						
2.3 Titularul activităților aplicative ⁶	Sl. dr. ing. Răzvan Aciu						
2.4 Anul de studii ⁷	I	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei ⁸	DI

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

3.1 Număr de ore asistate integral/săptămână	5 , format din:	3.2 ore curs	2	3.3 ore seminar /laborator /proiect	2/1
3.1* Număr total de ore asistate integral/sem.	70 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	28/14
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	5.71 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1.71
		ore studiu individual după manual, suport de curs, bibliografie și notițe			2
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			2
3.7* Număr total de ore activități neasistate/semestru	80 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			24
		ore studiu individual după manual, suport de curs, bibliografie și notițe			28
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			28
3.8 Total ore/săptămână ¹⁰	10.71				
3.8* Total ore/semestru	150				
3.9 Număr de credite	6				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor
4.2 de competențe	• Cunoștințe de matematică • Cunoștințe de programare

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

² Se înscrie numele facultății care gestionează programul de studii căruia îi aparține disciplina.

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

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

⁵ 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).

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

⁷ Anul de studii în care este prevăzută disciplina în planul de învățământ.

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

⁹ 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.

¹⁰ 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	• Sală mare, Materiale suport: laptop, proiector, tablă.
5.2 de desfășurare a activităților practice	• Laborator cu 15-20 calculatoare – Mediu de programare pentru limbajul C, tablă

6. Competențe la formarea cărora contribuie disciplina

Competențe specifice	<ul style="list-style-type: none"> • Dobândirea tehnicilor de programare avansată • Proiectarea și implementarea unor programe C de complexitate medie și mare • Însușirea unui stil de programare corect
Competențele profesionale în care se înscriu competențele specifice	C2 Proiectarea componentelor hardware, software și de comunicații C3 Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor C4 Proiectarea și integrarea sistemelor informatice utilizând tehnologii și medii de programare
Competențele transversale în care se înscriu competențele specifice	CT1 Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei CT3 Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională

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

7.1 Obiectivul general al disciplinei	• Completarea cunoștințelor de programare dobândite la cursul de inițiere în programare cu alte facilități ale limbajului C și cu elemente de tehnici de programare specifice unui curs de programare avansată
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Dobândirea tehnicilor de programare avansată • Proiectarea și implementarea unor programe C de complexitate medie și mare • Însușirea unui stil de programare corect

8. Conținuturi¹¹

8.1 Curs	Număr de ore	Metode de predare ¹²
1. Elemente de programare avansată în limbajul C 1.1 Structuri de tip union 1.2 Câmpuri de biți 1.3 Tabelul complet al precedentei operatorilor 1.4 Argumente în linia de comandă 1.5 Argumente de tip pointer 1.6 Pointeri la funcții 1.7 Preprocesorul 1.8 Funcții cu număr variabil de argumente	4	Prelegere susținută de prezentări PPT, conversații, explicații, exemplificări pe tablă. Suportul de curs este pe Internet
2. Fișiere 2.1 Caracteristici ale fișierelor 2.2 Operații de bază asupra fișierelor 2.3 Prelucrarea fișierelor text	4	

¹¹ 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, tematica fiecărui stagi de practică). Titlurile lucrărilor de laborator care se efectuează pe standuri vor fi însoțite de notația „(*)”.

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

2.4 Prelucrarea fişierelor binare 2.5 Funcţii speciale pentru tratarea erorilor 2.6 Actualizarea fişierelor în acces direct 2.7 Redirecţionarea intrării şi a ieşirii standard		
3. Proiectarea şi dezvoltarea sistematică a programelor de mari dimensiuni 3.1 Programarea calculatoarelor – de la teorie la practică 3.2 Stilul de programare 3.3 Etica programării 3.4 Metoda detaliilor în paşi succesivi 3.5 Exemplu de program 3.6 Concluzii	2	
4. Recursivitatea în C 4.1 Conceptul de recursivitate 4.2 Recursivitate directă 4.3 Înregistrarea de activare 4.4 Relaţia dintre recursivitate şi iteraţie 4.5 Exemple de programe recursive	2	
5. Metode generale de proiectare a algoritmilor şi programelor 5.1 Metoda Greedy 5.2 Metoda Backtracking 5.3 Metoda Divide and Conquer	4	
6. Pointeri. Alocarea dinamică a memorie 6.1 Pointeri şi adrese 6.2 Pointeri şi tablouri 6.3 Aritmetica adreselor 6.4 Pointeri spre caractere 6.5 Alocarea dinamică de memorie 6.6 Tablouri de pointeri; pointeri la pointeri	2	
7. Structuri de date dinamice 7.1 Structuri de date statice şi dinamice 7.2. Tipul pointer; utilizarea pointerilor 7.3 Pointeri structuri şi tablouri 7.4 Alocarea dinamică a memoriei 7.5 Liste liniare simplu înlanţuite 7.6 Liste ordonate 7.7 Liste dublu înlanţuite 7.8 Liste multiplu înlanţuite	4	
8. Tipuri de date abstracte 8.1 Crearea unui executabil din mai multe fişiere sursă 8.2. Fişiere antet 8.3. Variabile externe 8.4. Variabile şi funcţii statice 8.5. Tipuri de date abstracte	4	
9. Tehnici de căutare şi sortare 10.1 Principiile sortării 10.2 Sortare prin interschimbare 10.3 Sortare prin inserţie 10.4 Sortare prin partiţionare 10.5 Concluzii	2	
1. Bibliografie ¹³ Brian W. Kernighan, Denis Ritchie, <i>Limbajul C</i> , Ed. Teora, 2003 2. Horia Ciocârlie, Rodica Ciocârlie, <i>Tehnici de programare şi structuri de date</i> , Ed. Eurostampa, 2012 3. Dr. Kris Jamsa & Lars Klander, <i>Totul despre C şi C++</i> . Manualul fundamental de programare în C şi C++, Ed. Teora, 2009 4. Liviu Negrescu, <i>Limbajele C şi C++ pentru începători, vol. I, Limbajul C</i> , Ed. MicroInformatica, 1994 5. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, <i>Introducere în algoritmi</i> , Ed. Computer LIBRIS Agora, (MIT Press, 1990) 6. Donald E. Knuth, <i>Arta programării calculatoarelor (3 volume), Algoritmi fundamentali, Algoritmi seminumerici, Sortare şi căutare</i> , Ed. Teora, 2000, (Addison Wesley, 1962) 7. Robert Sedgewick, <i>Algorithms</i> , Ed. Addison Wesley, 1983.		

¹³ Cel puţin un un titlu trebuie să aparţină colectivului disciplinei iar cel puţin un titlu trebuie să se refere la o lucrare de referinţă pentru disciplină, de circulaţie naţională şi internaţională, existentă în biblioteca UPT.

8. V.Iorga, P.Chiriță, C. Stratan, C.Opincaru, <i>Programare în C/C++</i> . <i>Culegere de probleme</i> , Ed. Niculescu, 2003 9. Vladimir Crețu, <i>Structuri de date și algoritmi, vol. I, Structuri de date fundamentale</i> , Ed. Orizonturi Universitare, 2000 Ioana Șora, Doru Todinca, <i>Introducere în programarea calculatoarelor</i> , Ed. Politehnica, 2004		
8.2 Activități aplicative ¹⁴	Număr de ore	Metode de predare
8.2.1 LABORATOR		Expunere temă, discuții, întrebări, explicații la tablă, rezolvare pe calculator, a 1-2 probleme. Temele sunt pe Campus Virtual
1. Structuri de date avansate	2	
2. Proiectarea și dezvoltarea sistematică a programelor de mari dimensiuni	4	
3. Pointeri – noțiuni avansate		
4. Prelucrări de fișiere	4	
5. Recursivitate	2	
6. Aplicații ale metodei Greedy	6	
7. Algoritmi cu revenire (Backtracking)		
8. Liste simplu înlănțuite, operații specifice, metoda celor doi pointeri și a fanionului	4	
9. Liste dublu înlănțuite, operații specifice		
10. Aplicații cu date abstracte	2	
11. Aplicații ale tehnicilor de sortare	2	
12. Recuperări	2	
8.2.2 PROIECT	14	Fiecare grupă de 2-3 studenți primește o temă de proiect, încadrată într-una dintre cele 11 tematici prezentate la conținutul laboratorului Susținerea finală se face pe baza unei prezentări PPT
1. Studierea temei primite și a variantelor de implementare	2	
2. Proiectarea soluției	4	
3. Scrierea programelor	2	
4. Testarea și validarea programelor. Elaborarea documentației de realizare	4	
5. Susținerea proiectului	2	
1. Bibliografie ¹⁵ Brian W. Kernighan, Denis Ritchie, <i>Limbaajul C</i> , Ed. Teora, 2003 2. Horia Ciocârlie, Rodica Ciocârlie, <i>Tehnici de programare și structuri de date</i> , Ed. Eurostampa, 2012 3. Dr. Kris Jamsa & Lars Klander, <i>Totul despre C și C++</i> . <i>Manualul fundamental de programare în C și C++</i> , Ed. Teora, 2009 4. Liviu Negrescu, <i>Limbaajele C și C++ pentru începători, vol. I, Limbaajul C</i> , Ed. MicroInformatica, 1994 5. Thomas H. Cormen, Charles E. Leiserson, Ronald R. Rivest, <i>Introducere în algoritmi</i> , Ed. Computer LIBRIS Agora, (MIT Press, 1990) 6. Donald E. Knuth, <i>Arta programării calculatoarelor (3 volume), Algoritmi fundamentali, Algoritmi seminumerici, Sortare și căutare</i> , Ed. Teora, 2000, (Addison Wesley, 1962) 7. Robert Sedgewick, <i>Algorithms</i> , Ed. Addison Wesley, 1983. 8. V.Iorga, P.Chiriță, C. Stratan, C.Opincaru, <i>Programare în C/C++</i> . <i>Culegere de probleme</i> , Ed. Niculescu, 2003		

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

- Cunoștințele de programare sunt importante pentru toate materiile cu specific software care fac parte din planul de învățământ al specializării: Tehnici de programare, Programare orientată pe obiecte, Structuri de date și algoritmi, Proiectarea și analiza algoritmilor, Fundamentele ingineriei software, Sisteme de operare ș. a.
- Majoritatea angajatorilor reprezentativi din domeniul aferent programului solicită atât cunoștințe de programare în general cât și cunoașterea limbajului de programare C.

10. Evaluare

¹⁴ 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:”.

¹⁵ Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

Tip activitate	10.1 Criterii de evaluare ¹⁶	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Rezolvarea a doua probleme din prima parte și respectiv din a doua parte a materiei Stil de programare	Examinare orală, pe calculator	55 %
		Examinare orală, pe calculator	5 %
10.5 Activități aplicative	S:		
	L: Rezolvarea problemelor corespunzătoare lucrărilor de laborator Teme de casă	Prezentarea rezolvărilor, răspunsuri la întrebări	20 %
		Prezentarea rezolvărilor, răspunsuri la întrebări	5 %
	P¹⁷: Fiecare grupă de 2-3 studenți susține tema de proiect primită	Susținerea se face pe baza unei prezentări PPT	15 %
	Pr:		
10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁸)			
<ul style="list-style-type: none"> Proiectarea, testarea și executarea unui program de complexitate medie din conținutul materiei Cele două programe de pe biletul de examen trebuie să fie funcționale și să rezolve minimul de cerințe solicitat 			

Data completării

5.06.2019

**Titular de curs
(semnătura)**

.....

**Titular activități aplicative
(semnătura)**

.....

**Director de departament
(semnătura)**

.....

Data avizării în Consiliul Facultății¹⁹

16.10.2019

**Decan
(semnătura)**

.....

¹⁶ 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.)

¹⁷ 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.

¹⁸ Nu se va explica cum se acorda nota de promovare.

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