

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN ORADEA
1.2 Facultatea	FACULTATEA DE INGINERIE ELECTRICĂ ȘI TEHNOLOGIA INFORMAȚIEI
1.3 Departamentul	INGINERIA SISTEMELOR AUTOMATE ȘI MANAGEMENT
1.4 Domeniul de studii	INGINERIA SISTEMELOR
1.5 Ciclul de studii	LICENȚĂ
1.6 Programul de studii/Calificarea	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ /INGINER

2. Date despre disciplină

2.1 Denumirea disciplinei	Programarea calculatoarelor și limbaje de programare II						
2.2 Titularul activităților de curs	Sef.lucr.dr.ing. Coman Simina						
2.3 Titularul activităților de laborator/proiect	Sef.lucr.dr.ing. Mesaros Diana						
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare	Ex	2.7 Regimul disciplinei	DF

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 laborator /proiect	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 laborator /proiect	28
Distribuția fondului de timp ore					69 ore
Studiul după manual, suport de curs, bibliografie și notițe					28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					18
Tutoriat					2
Examinări					6
Alte activități.....					
3.7 Total ore studiu individual	69				
3.9 Total ore pe semestru	125				
3.10 Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de competențe	Competențe de programare în limbajul C

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	- sală de curs dotată cu calculator și videoproiector – prezentare bazată pe slide-uri - prezență minim 50% din cursuri
5.2. de desfășurare a laboratorului /proiectului	- sală de laborator dotată cu calculatoare, C/C++ (Visual Studio/DevC++/MinGW) - prezența obligatorie la toate laboratoarele; - se pot recupera pe parcursul semestrului maxim 4 lucrări (30 %); - frecvența la orele de laborator sub 70% conduce la refacerea disciplinei

6.1. Competențele specifice acumulate	
Competențe profesionale	C2. Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor
Competențe transversale	CT2. Identificarea rolurilor și responsabilităților într-o echipă plurispecializată luarea deciziilor și atribuirea de sarcini, cu aplicarea de tehnici de relaționare și muncă eficientă în cadrul echipei CT3. Identificarea oportunităților de formare continuă și valorificarea eficienței a resurselor și tehnicilor de învățare pentru propria dezvoltare.

6.2. Rezultatele așteptate ale învățării

Cunoștințe	2. Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale din sisteme automate, sisteme încorporate și inteligente, știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete.
Aptitudini	2. Studentul/absolventul utilizează limbaje, medii și tehnologii de programare și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) în rezolvarea de probleme bine definite din ingineria sistemelor. Studentul/absolventul evaluează și selectează, adaptează și extinde mijloacele de proiectare asistată de calculator (CAD), pachetele hardware și software dedicat, pentru implementarea sistemelor complexe conectate ingineriei sistemelor. Studentul/absolventul modelează la nivel de sistem și realizează sisteme de control numerice, interfețe și sisteme bazate pe microcontrolere și microprocesoare, folosind proiectarea hardware – software integrată (co-design) și ingineria programării.
Responsabilitate și autonomie	2. Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.

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

7.1 Obiectivul general al disciplinei	▪ Continuând elementele de programare începute în semestrul anterior, cursul își propune familiarizarea studenților cu o serie de tehnici și concepte de programare avansate care permit proiectarea și dezvoltarea unor programe cu un grad ridicat de complexitate ▪ Cursul are un caracter puternic aplicativ, în scopul aprofundării depinderilor practice de programare ale studenților, conținând un număr mare de exemple de algoritmi în format sursă, fără a restrânge însă generalitatea conceptelor prezentate.
7.2 Obiectivele specifice	▪ Cursul își propune prezentarea unor tehnici și concepte de programare avansate împreună cu metode și algoritmi specifici, care permit proiectarea și implementarea unor programe complexe, în scopul rezolvării diferitelor tipuri de aplicații: manipulare avansată a tablourilor, fișierelor, șirurilor de caractere, împreună cu o serie de algoritmi cunoscuți în domeniu. Un capitol separat abordează, prin exemple, problema

	recursivității. Sunt prezentate, de asemenea, noțiuni fundamentale de evaluare a performanțelor algoritmilor, exemplificate prin evaluări comparative precum și de proiectare și implementare a programelor complexe ▪ Laboratorul, realizat utilizând limbajul C, familiarizează studenții cu aspecte practice privind rezolvarea diferitelor tipuri de probleme prin implementarea și adaptarea unor algoritmi și tipuri de date specifice
--	---

8. Conținuturi

8.1.Curs	Metode de predare	Observații
CAP. 1. INTRODUCERE 1.1. Tipuri și structuri de date 1.1.1. Tipuri de date statice 1.1.2. Tipuri de date dinamice 1.2. Evaluarea performanțelor algoritmilor – concepte utilizate, notația $O(n)$	Expunere liberă, cu prezentarea cursului pe videoproiector și pe tablă	2 h
CAP. 2. TIPUL DE DATE TABLOU 2.1. Concepte avansate de manipulare a tablourilor 2.1.1. Tablouri unidimensionale/multidimensionale 2.1.2. Manipularea tablourilor, aritmetica pointerilor, indcsi, managementul memoriei 2.2. Algoritmi de căutare în tablouri 2.2.1. Tehnica fanionului 2.2.2. Căutarea binară 2.3. Algoritmi pentru sortarea tablourilor 2.3.1. Metode directe de sortare 2.3.1.1. Tehnica sortării prin inserție 2.3.1.2. Tehnica sortării prin selecție 2.3.1.3. Tehnica sortării prin interschimbare 2.3.2. Metode avansate de sortare 2.3.2.1. Tehnica sortării prin inserție cu diminuarea incrementului (shellsort) 2.3.2.2. Tehnica sortării arborilor prin metodaansamblelor (heapsort) 2.3.2.3. Tehnica sortării prin partiționare (quicksort) 2.4. Utilizarea structurii de tablou în implementarea structurilor de tip LIFO și FIFO. Tablouri circulare.	Expunere liberă, cu prezentarea cursului pe videoproiector și pe tablă	2h 2h 2h
CAP. 3. TIPUL DE DATE FIȘIER 3.1. Concepte avansate de manipulare a fișierelor 3.2. Algorimi pentru sortarea fișierelor secvențiale 3.2.1. Tehnica sortării fișierelor prin interclasare echilibrată 3.2.2. Tehnica sortării fișierelor prin interclasare naturală	Expunere liberă, cu prezentarea cursului pe videoproiector și pe tablă	2h 2h
CAP 4.TIPUL DE DATE ȘIR 4.1. Concepte avansate de manipulare a șirurilor 4.2. Algoritmi de căutare în șiruri	Expunere liberă, cu prezentarea cursului pe videoproiector și pe tablă	2h 2h
CAP. 5. RECURSIVITATE 5.1. Generalități. 5.2. Algoritmi recursivi. Exemple 5.2.1. Algoritmi de divizare 5.2.2. Algoritmi recursivi pentru determinarea tuturor soluțiilor unor probleme 5.2.3. Algoritmi cu revenire (backtracking) 5.2.4. Algoritmi pentru determinarea optimului (knapsack) 5.3. Recursivitate și structuri de date 6.3.1. Structuri cu autoreferire: liste înlănțuite, liste dublu înlănțuite, structura de multilistă	Expunere liberă, cu prezentarea cursului pe videoproiector și pe tablă	2h 2h 2h 2h
CAP. 6. TEHNICA DISPERSIEI 6.1. Principiul tehnicii dispersiei 6.2. Determinarea funcției de dispersie. Tratarea situației de coliziune	Expunere liberă, cu prezentarea cursului pe videoproiector și pe tablă	2h
Bibliografie 1. http://www.cprogramming.com/ 2. http://www.algolist.net/Algorithms/		

3. P.J.Deitel, H.M. Deitel, <i>C: How to program</i> , Pearson Education International, ISBN 0-13-239300-X, Fifth Edition, 2007 4. D. Knuth, <i>Arta programarii calculatoarelor</i> , volumul 3 - Sortare si cautare, Editura Teora, 2004 5. D. Zmaranda - <i>Algoritmi și tehnici de programare</i> , Editura Universității din Oradea, ISBN 973-613-062-2, 264 pg., 2001, versiune electronică actualizată 2014, https://uoradea-my.sharepoint.com/personal/rodica_zmaranda_didactic_uoradea_ro/Documents/PCLPII/PCLP_II.pdf 6.V. Crețu, <i>Structuri de date și algoritmi – vol. 1: Structuri de date fundamentale</i> , Editura Orizonturi Universitare Timisoara, ISBN 973-9400-74-4, 2000 7.Coman Simina, <i>Programarea calculatoarelor și limbaje de programare</i> , curs în format electronic, 2021		
8.2. Laborator	Metode de predare	Observații
1. Determinarea experimentală a timpului de execuție al unui program	Studentii primesc temele pentru laborator cu cel puțin o săptămână înainte,	2 h
2. Tablouri. Tehnici de căutare în tablouri.	și le studiază (problemele de la sfârșitul	2 h
3. Tehnici de sortare directe ale tablourilor	laboratorului).	2 h
4. Tehnici de sortare avansate ale tablourilor	La începutul laboratorului se discută modalitățile de	2 h
5. Utilizarea structurii de tablou. Tablouri circulare	rezolvare a aplicațiilor propuse. Pe urmă, studenții realizează partea practică a	2 h
6. Fișiere. Sortarea fișierelor	lucrării (problemele propuse) sub îndrumarea	4 h
7. Tehnici de cautare în siruri de caractere	cadrului didactic.	4 h
8. Recursivitate – algoritmi recursivi		2 h
9. Recursivitate - structuri cu autoreferire		2 h
10. Tehnica dispersiei		2 h
11. Predarea lucrărilor, încheierea situației la laborator		2 h
12. Recuperări		2 h
Bibliografie 1. D. Zmaranda , Coman Simina - <i>Algoritmi și tehnici de programare</i> – îndrumător de laborator, volumul I , Editura Universității din Oradea, ISBN 973-613-302-8, 100 pg., 2003, versiune electronică actualizată 2014, https://uoradea-my.sharepoint.com/personal/rodica_zmaranda_didactic_uoradea_ro/Documents/PCLPII/Laborator_PCLPII.pdf		

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

- Conținutul disciplinei se regăsește în curricula specializărilor similare și din alte centre universitare care au acreditat aceste specializări (Universitatea „Politehnica” din Timișoara, Politehnica București) iar cunoașterea tipurilor de date și a algoritmilor prezentați în cadrul acestei discipline reprezintă o cerință fundamentală în scopul formării deprinderilor și abilităților de bază de programare necesare

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- pentru nota 5 sunt necesare răspunsuri corecte la minim 8 subiecte tip grilă - pentru nota 10, este necesară răspunsul corect la toate subiectele din cadrul grilei	Examen scris Studentii primesc spre rezolvare fiecare câte un formular cu 18 subiecte de teorie, tip grilă	50 %
10.5 Laborator	- pentru nota 5, realizarea unei implementări funcționale în proporție de 50% a problemelor propuse la laborator - pentru nota 10, cunoașterea amănunțită a modalității de implementare practică a tuturor lucrărilor de laborator	Aplicație practică La fiecare laborator studenții sunt evaluați pe baza activității (răspunsuri la întrebări, propuneri de implementare, etc.), evaluări care se concretizează la finalul laboratorului într-o notă pe activitatea de laborator din cursul semestrului. De asemenea, în ultima oră de laborator studenții finalizează și predau toate aplicațiile practice propuse în cadrul laboratorului. Media dintre nota primită pentru aplicațiile practice și nota de la activitatea de la laborator va reprezenta nota finală la laborator	50%

10.7 Standard minim de performanță
• cunoașterea modalității de evaluare analitică a performanțelor unui algoritm, evaluarea comparativă a performanțelor unor algoritmi simpli • cunoașterea și înțelegerea tehnicilor de programare utilizate în manipularea tablourilor și fișierelor, în cadrul metodelor de căutare în tablouri precum și a metodelor sortare directă și avansată a tablourilor și a fișierelor; aplicarea tehnicilor și conceptelor în diverse categorii de aplicații • înțelegerea mecanismului recursivității, familiarizarea cu principalele tipuri de algoritmi recursivi și aplicarea diverselor tipuri de algoritmi recursivi în cadrul unor aplicații specifice; manipularea structurilor cu autoreferire și utilizarea acestora în cadrul aplicațiilor • cunoașterea modalităților de manipulare a șirurilor și a algoritmilor specifici de căutare în șiruri și utilizarea acestora în aplicații specifice

Data completării

02.09.2025

Semnătura titularului de curs,

Sef.lucr.dr.ing. Coman Simina
ccoman@uoradea.ro

Semnătura titularului de
seminar/laborator/proiect,
Ș.l. dr. ing. Mesaros Diana
dsas@uoradea.ro

Data avizării în departament

18.09.2025

Semnătura directorului de departament

Prof.univ.dr.ing. Helga Silaghi
e-mail: hsilaghi@uoradea.ro

Data avizării în Consiliul facultății

25.09.2025

Semnătură Decan
Conf.dr. ing. Eugen Gergely
e-mail: egergely@uoradea.ro