



FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea din Craiova
1.2. Facultatea	Automatică, Calculatoare și Electronică
1.3. Departamentul	AUTOMATICĂ ȘI ELECTRONICĂ (D28)
1.4. Domeniul de studii	Automatică, informatică aplicată și sisteme inteligente
1.5. Ciclul de studii universitare	LICENȚĂ
1.6. Forma de organizare	IF (Învățământ cu frecvență)
1.7. Programul de studii	AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ (cod D28AIAL204)

2. Date despre disciplină

2.1. Denumirea disciplinei			Ingineria Sistemelor de Programe				
2.2. Titularul activităților de curs			Ș.l. dr. ing. Laviniu Aurelian BĂDULESCU				
2.3. Titularul activităților de seminar/ laborator			Ș.l. dr. ing. Bogdan POPA				
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DOB

3. Timpul total estimat (ore pe semestru a activităților didactice)

3.1. Numărul de ore pe săptămână	4	din care: 3.2 curs	2	3.3. seminar/laborator	2
3.4. Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6. seminar/laborator	28
Distribuția fondului de timp - ore/sapt.					8,93
Studiul după manual, suport de curs, bibliografie și notițe					35
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					11
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					11
Tutoriat					3
Examinări					3
Alte activități: consultații, cercuri studențești					6
3.7. Total ore studiu individual					69
3.8. Total ore pe semestru					125
3.9. Numărul de credite					5

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Studentul trebuie să posede cunoștințe de specialitate dobândite la următoarele discipline: Programarea calculatoarelor și limbaje de programare, Analiză matematică, Matematici speciale.
4.2. de competențe	Nu sunt necesare.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Predarea cursului se face folosind videoproiectorul. Pentru explicații mai detaliate ale conceptelor ilustrate în diapozitivele care constituie suportul de curs și pentru a răspunde întrebărilor formulate de studenții din sală se folosește tabla. În cazul predării online se folosesc facilitățile aplicației
--------------------------------	---

	Google Classroom. Se asigură suport de curs în format electronic și acces la documentații actualizate. Procesul de predare are următoarea structură: • 80% prezentare teoretică, pe baza suportului de curs (diapozitive) • 20% activitate interactivă (discuții cu studenții)
5.2. de desfășurare a seminarului/ laboratorului	Laboratorul utilizează o rețea de calculatoare. Sunt folosite calculatorul și mediile de programare vizuală pentru exemplificarea conceptelor prezentate la curs, precum și pentru rezolvarea unor tipuri de probleme propuse studenților la laborator. În cazul predării online se folosesc facilitățile aplicației Google Classroom.

6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

Cunoștințe	Studentul/Absolventul: 1. descrie concepte fundamentale din știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete. 2. identifică concepte fundamentale din știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete. 3. sumarizează concepte fundamentale din știința calculatoarelor și tehnologia informației și modul lor de aplicare în probleme concrete.
Aptitudini (Abilități)	Studentul/Absolventul: 1. utilizează limbaje, medii și tehnologii de programare și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) în rezolvarea de probleme bine definite din ingineria sistemelor. 2. evaluează și selectează, adaptează și extinde pachete software dedicat, pentru implementarea sistemelor complexe conectate ingineriei sistemelor. 3. modelează la nivel de sistem și realizează sisteme folosind ingineria programării.
Responsabilitate și autonomie	Studentul/Absolventul: 1. arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.

7. Conținuturi

7.1. CURS	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
1 Analiza complexității algoritmilor <i>1.1. Introducere; 1.2. Analiza complexității algoritmilor.</i>	față în față	Predarea cursului se face folosind video-proiector. Explicațiile suplimentare sau detaliate se fac folosind tabla. Exemplificarea practică a conceptelor prezentate și rezolvarea problemelor se face direct în mediul vizual de programare. În cazul predării online	2
2 Recursivitatea <i>2.1. Recursivitatea; 2.2. Tehnica de programare divide et impera.</i>	față în față		2
3 Căutarea <i>3.1. Căutarea secvențială; 3.2. Căutarea cu santinelă; 3.3. Căutarea într-un vector ordonat; 3.4. Căutarea binară; 3.5. Căutarea binară uniformă; 3.6. Căutarea prin interpolare.</i>	față în față		2
4 Selecția <i>4.1. Selecția maximului; 4.2. Selecția minimului și maximului; 4.3. Selecția celor mai mari k elemente; 4.4. Selecția k-elementului.</i>	față în față		2
5 Sortarea (1) <i>5.1. Sortarea cu metoda bulelor; 5.2. Sortarea rapidă; 5.3. Sortarea prin selecție directă.</i>	față în față		2
6 Sortarea (2) <i>6.1. Sortarea Heap; 6.2. Sortarea prin inserție directă; 6.3. Sortarea Shell.</i>	față în față		2
7 Sortarea (3)	față în față		2

7.1. Sortarea prin fuziune; 7.2. Sortarea prin numărare; 7.3. Sortarea prin numărarea distribuțiilor.		se folosesc facilitățile aplicației Google Classroom.	
8 Tehnica de programare backtracking (1) 8.1. Problema reginelor; 8.2. Structura unei implementări; 8.3. Problema colorării hârtilor.	față în față		2
9 Tehnica de programare backtracking (2) 9.1. Problema comis-voiajorului; 9.2. Problema labirintului; 9.3. Problema săriturii calului.	față în față	Cursul este structurat în raportul de mai jos:	2
10 Tehnica de programare Greedy (1) 10.1. Problema planificării spectacolelor; 10.2. Problema rucsacului.	față în față	- 80% prezentare teoretică, pe baza suportului de curs	2
11 Tehnica de programare Greedy (2) 11.1. Problema săriturii calului; 11.2. Problema comis-voiajorului.	față în față	(diapozitive). - 20% activitate interactivă (discuții cu studenții)	2
12 Tehnica programării dinamice 12.1. Problema triunghiului; 12.2. Problema subșirului crescător de lungime maximă. 12.3. Problema rucsacului; 12.4. Greedy, backtracking sau programare dinamică.	față în față	Materialele necesare vor fi puse la dispoziția studenților în format electronic.	2
13 Generarea de combinații, permutări și aranjamente 13.1. Generarea permutărilor; 13.2. Generarea combinațiilor; 13.3. Generarea aranjamentelor.	față în față		2
14 Grafuri și arbori 14.1. Grafuri orientate și neorientate; 14.2 Arbori	față în față		2
Bibliografie:			
1. Cormen, T., H., Leiserson, C., E., Rivest, R., R., <i>Introducere în algoritmi</i> , Ed. Computer Libris Agora, Cluj-Napoca, 2000.			
2. Knuth, D., <i>Arta Programării Calculatoarelor: Vol. 1 Algoritmi Fundamentali</i> , Ed. Teora, București, 1999.			
3. Knuth, D., <i>Arta Programării Calculatoarelor: Vol. 2 Algoritmi Seminumerici</i> , Ed. Teora, București, 2000.			
4. Knuth, D., <i>Arta Programării Calculatoarelor: Vol. 3 Sortare și Căutare</i> , Ed. Teora, București, 2001.			
5. Bădulescu, L., A., <i>Ingineria Sistemelor de Programe</i> , manual de curs, 2025.			

7.2. Seminar/laborator	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
Analiza complexității algoritmilor.	față în față	Efectuarea lucrărilor de laborator se face individual de fiecare student folosind exemple de probleme rezolvate, teme propuse spre rezolvare și utilizând mediul de dezvoltare vizuală disponibil pe calculatorul din rețeaua de laborator. În cazul predării online se folosesc facilitățile aplicației Google Classroom.	2
Recursivitatea; Tehnica de programare divide et impera.	față în față		2
Aplicații de căutare.	față în față		2
Aplicații de selecție	față în față		2
Aplicații de sortare cu metoda bulelor, quicksort și selecție directă.	față în față		2
Aplicații de sortare cu Heap, prin inserție directă și Shellsort.	față în față		2
Aplicații de sortare prin fuziune, prin numărare și prin numărarea distribuțiilor.	față în față		2
Aplicații backtracking: problema reginelor, problema colorării hârtilor.	față în față	Sunt puse la dispoziția studenților platforme de laborator care conțin un breviar teoretic și modul de desfășurare al lucrării.	2
Aplicații backtracking: problema comis-voiajorului, problema labirintului și problema săriturii calului.	față în față		2
Aplicații Greedy: problema planificării spectacolelor și problema rucsacului.	față în față		2
Aplicații Greedy: problema săriturii calului și problema comis-voiajorului.	față în față	Activități prevăzute: • 50% desfășurarea lucrării	2
Aplicații cu programarea dinamică: problema triunghiului, problema subșirului crescător de	față în față		2

lungime maximă și problema rucsacului.		50% interpretarea rezultatelor și discuții cu studenții	
Aplicații cu generarea permutărilor, generarea combinărilor și generarea aranjamentelor.	față în față		2
Grafuri și arbori	față în față		2
Bibliografie:			
1. Cormen, T., H., Leiserson, C., E., Rivest, R., R., <i>Introducere în algoritmi</i> , Ed. Computer Libris Agora, Cluj-Napoca, 2000.			
2. Knuth, D., <i>Arta Programării Calculatoarelor: Vol. 1 Algoritmi Fundamentali</i> , Ed. Teora, București, 1999.			
3. Knuth, D., <i>Arta Programării Calculatoarelor: Vol. 2 Algoritmi Seminumerici</i> , Ed. Teora, București, 2000.			
4. Knuth, D., <i>Arta Programării Calculatoarelor: Vol. 3 Sortare și Căutare</i> , Ed. Teora, București, 2001.			
5. Bădulescu, L., A., <i>Ingineria Sistemelor de Programe</i> , platforme de laborator, 2025.			

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

- QUEST GLOBAL ENGINEERING SERVICES Craiova
- Forvia Craiova
- CS România S.A.

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Înțelegerea fundamentelor teoretice corespunzătoare Ingineriei sistemelor de programe. Capacitatea de a realiza conexiuni între noțiunile predate. Capacitatea de analiză și sinteză într-o situație concretă.	Examen scris final tip grilă cu subiecte de programare	50%
9.5. Seminar/laborator	Capacitatea de analiză, identificare a soluției optime și rezolvare a problemelor propuse. Capacitatea de a aplica în practică noțiunile teoretice învățate.	Testări de laborator săptămânale pe parcursul semestrului și o testare finală.	40% Se acordă un punct din oficiu în nota finală.
9.6. Standard minim de performanță			
50% din rezultate așteptate ale învățării			

Data completării
01.10.2025

Titular de disciplină,
Ș.l. dr. ing. Laviniu Aurelian BĂDULESCU

Semnătura titularului
.....

Data avizării în departament
01.10.2025

Director de departament,
Conf. dr. ing. Cătălin CONSTANTINESCU

Semnătura directorului de departament,
.....

Decan,
Prof. dr. ing. Dorin ȘENDRESCU

Semnătura decanului,
.....