

FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Inginerie Electrică și Știința Calculatoarelor
Departamentul	Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licență, învățământ cu frecvență, dual
Programul de studii	Calculatoare

2. Date despre disciplină

Denumirea disciplinei	STRUCTURI DE DATE SI ALGORITMI				
Anul de studiu	I	Semestrul	2	Tipul de evaluare	E
Regimul disciplinei	Categoriza formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categoriza de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				DOB

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	Total general	5	Cur s	3	Semina r	Laborator IIS	2	Proiect IIS	Practic ă IIS
I b) Totalul de ore pe semestru din planul de învățământ		70	Cur s	4 2	Semina r	Laborator	28	Proiect	Practic ă

Distribuția fondului de timp pe semestru	Ore IIS	Ore IM
II.a) Studiu individual	30	72
II.b) Tutoriat (pentru ID)		
III. Examinări	3	
IV. Alte activități (precizați):		

Total ore studiu individual (II.a+II.b+III)	Ore IIS	33	Ore IM	72
Total ore pe semestru (Ib+II+III+IV)	Ore IIS	75	Ore IM	100
Numărul de credite	Credite IIS	3	Credite IM	4

4. Competențe specifice acumulate

Competențe profesionale/generale	C04. definește arhitectura software C05. creează softuri C06. definește cerințe tehnice C07. dezvoltă prototipul pentru software C16. utilizează biblioteci de software C19. identifică cerințele clienților
Competențe transversale	CT1. munca în echipă

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/absolventul dobândește cunoștințe fundamentale despre structuri de date și algoritmi în Python și este capabil să proiecteze, implementeze și analizeze soluții algoritmice eficiente, utilizând tipuri de date abstracte și principii de programare modulară	Studentul/absolventul specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbajul Python, aplicând principii de programare structurată și modulară, tehnici algoritmice (recursivitate, Greedy, Backtracking) și bune practici de proiectare software.	Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

	Studentul/absolventul alege, explică și aplică concepte specifice structurilor de date fundamentale (liste, stive, cozi, arbori, grafuri, tabele de dispersie), utilizând limbajul Python pentru implementarea și evaluarea soluțiilor eficiente.	
--	---	--

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

Obiectivul general al disciplinei	Formarea competențelor avansate de programare în Python prin aplicarea metodelor Greedy și Backtracking, utilizarea structurilor de date fundamentale și dezvoltarea programelor structurate și modularizate.
-----------------------------------	---

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observații
I. Introducere în limbajul Python		expunere orală, conversație, exemplu demonstrativ, sinteza cunoștințelor, descoperire dirijată, exemplificare, studiu de caz	
1. Tipuri de date. Operatori. Variabile locale, globale. Tratarea excepțiilor. Structuri de control.	4		
2. Definirea funcțiilor în Python. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor). Recursivitate în Python. Aplicații	4		
3. Manipularea șirurilor de caractere în Python	2		
4. Containere în Python (liste, dicționare, multimi, tuple)	4		
5. Programare modulară. Operații I/O în Python	3		
II. Introducere în TDA (tipuri de date abstracte)			
1. Abstractizarea și încapsularea datelor. Tipuri de date abstracte: domeniu, operații și reprezentarea datelor. Proiectarea TDA-urilor	3		
2. TDA Lista	3		
3. TDA stiva	3		
4. TDA coadă	2		
5. TDA arbore	2		
III. Grafuri în Python	3		
IV. Tabele de dispersie	2		
V. Tehnici de programare			
1. Metoda Greedy. Prezentare metodă. Aplicații	3		
2. Metoda Backtracking. Prezentare metoda. Aplicații	4		

Bibliografie minimală recomandată

1. Luciano Ramalho, Fluent Python. Clear, Concise, and Effective Programming, O'Reilly Media, August 2015.
2. Rance D. Necaise, Data Structures and Algorithms Using Python, Ed. JOHN WILEY & SONS, 2011.
3. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Data Structures and Algorithms in Python, John Wiley & Sons, 2013.
4. Python Software Foundation - <https://www.python.org/>, versiune electronică, 2025
5. John V. Guttag, Introduction to Computation and Programming Using Python. The MIT Press (2021)
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press (2022)
7. Stowell, Loue, Dickins, Rosie, Radulian, Cora, Programare pentru începători folosind Python. București : Litera, 2018.
8. Suport de curs in format electronic publicat pe Google Classroom

Aplicații (Seminar / laborator / lucrări practice / proiect)	Nr. ore	Metode de predare	Observații
1. Norme de protecția muncii. Introducere în Python. Prezentarea mediului de dezvoltare Python. Structuri de control în Python.	2	lucrări practice, dezvoltarea de programe pe calculator, experimentul	În cadrul lucrărilor de laborator se urmărește ca programele realizate de studenți să respecte o serie de reguli de bază ale ingineriei software. Temele de casă vor consta din dezvoltarea unor programe de
2. Manipularea șirurilor de caractere în Python	2		
3. Aplicații folosind colecții în Python (liste, tuple)	2		
4. Manipularea funcțiilor în Python	2		
5. Test de evaluare (1)	2		

6. Aplicații folosind dicționare și mulțimi în Python	2	complexitate medie, crearea documentației aferente produsului program realizat. Etape principale în dezvoltarea temei: • însușirea temei de realizat; • proiectarea logică a produsului program; • stabilirea structurilor de date și a algoritmului; • inserarea de comentarii în codul sursă • punerea la punct a produsului program; • testarea cu date impuse.
7. Operații cu fișiere text.	2	
8. Implementarea TDA listă utilizând limbajul Python	2	
9. Implementarea TDA stivă și coada utilizând limbajul Python	2	
10. Aplicații TDA arbori	2	
11. Aplicații grafuri	2	
12. Aplicații metoda Greedy.	2	
13. Aplicații metoda Backtracking.	2	
14. Test de evaluare (2)	2	
Prezența la activitățile obligatorii este reglementată de “R12. Regulamentul cadru privind evaluarea studenților” (https://usv.ro/fisiere_utilizator/file/ceac/regulamente/R12%20Reg%20cadru%20privind%20evaluarea%20stud_Ed%204rev2_25.05.2023_site.pdf). Recuperarea activităților aplicative obligatorii programate pe parcursul semestrului se face în conformitate cu precizările aceluiași regulament. De asemenea, în conformitate cu prevederile articolului 5.2.3, activitățile pe parcurs pot fi echivalate cu proiecte din tematica disciplinei cu acordul cadrului didactic titular.		
Bibliografie minimală recomandată 1. Luciano Ramalho, <i>Fluent Python</i>. Clear, Concise, and Effective Programming, O'Reilly Media, August 2015. 2. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, <i>Data Structures and Algorithms in Python</i>, John Wiley & Sons, 2013. 3. Python Software Foundation - https://www.python.org/, versiune electronică, 2022. 4. Tutoriale Python: http://docs.python.org/tutorial/index.html, https://www.codecademy.com/learn/learn-python, versiune electronică, 2025 7. Stowell, Loue, Dickins, Rosie, Radulian, Cora, <i>Programare pentru începători folosind Python</i>. București : Litera, 2018. 8. Lucrări de laborator în format electronic publicat pe Google Classroom		

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	• Cunoașterea noțiunilor teoretice privind structurile de date fundamentale (liste, stive, cozi, mulțimi, arbori); • Înțelegerea metodelor de rezolvare a problemelor (Greedy, Backtracking); • Capacitatea de a explica concepte de proiectare structurată și modularizată a programelor; • Claritatea, coerența și rigoarea exprimării în răspunsuri.	Evaluare prin probă finală mixta (proba scrisă și proba practică pe calculator)	50%
Seminar			
Laborator/ Lucrări practice	• Corectitudinea și complexitatea soluțiilor implementate; • Utilizarea adecvată a structurilor de date și a metodelor studiate.	Evaluare continuă (prin metode orale și probe practice)	50%
Proiect			

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
25.09.2025	Ș.I. dr. ing. Ionela RUSU	drd. ing. Adrian Vasile CATANĂ

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
25.09.2025	Prof. dr. ing. Cristina Elena TURCU

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Prof. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
26.09.2025	Prof. dr. ing. Laurentiu- Dan MILICI