FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul	Calculatoare, Electronică şi Automatică
Domeniul de studii	Automatică, informatică aplicată şi sisteme inteligente
Ciclul de studii	Licenţă, dual
Programul de studii	Automatică şi informatică aplicată

2. Date despre disciplină

Denumirea disciplinei	PROGRAMAREA CALCULATOARELOR ȘI LIMBAJE DE PROGRAMARE II				
Anul de studiu	1	Semestrul	2	Tipul de evaluare	E
Regimul disciplinei	Categoria formativă a disciplinei DF – fundamentală, DS – de specializare, DC – complementară				DF
	Categoria 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	4	Curs	2	Seminar		Laborator IIS	2	Proiect IIS		Practică IIS	
							Laborator IM		Proiect IM		Practică IM	
I b) Totalul de ore pe semestru din planul de învățământ		56	Curs	28	Seminar		Laborator	28	Proiect		Practică	

Distribuția fondului de timp pe semestru	ore
II. Studiu individual	41
III. Examinări	3
IV. Alte activități (precizați):	

Total ore studiu individual II	Ore IIS	44	Ore IM	
Total ore pe semestru (Ib+II+III+IV)	Ore IIS	100	Ore IM	
Numărul de credite	Credite IIS	4	Credite IM	

4. Competențe specifice acumulate

Competențe transversale	CT6. Își asumă responsabilitatea
Competențe profesionale/generale	CP9. Dezvolta aplicatii de procesare de date CP14. Utilizeaza interfete specifice aplicatiilor CP16. Creeaza softuri

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/absolventul explică și interpretează rezultate teoretice și experimentale din matematică, fizică, chimie, economie, desen tehnic și informatică.	- Studentul/absolventul aplică criterii și metode de evaluare pentru identificarea, modelarea, experimentarea, analiza și aprecierea calitativă și cantitativă a fenomenelor și proceselor specifice domeniului fundamental folosind inclusiv tehnologii digitale. - Studentul/absolventul achiziționează și prelucrează date, interpretează rezultate teoretice și experimentale. - Studentul/absolventul concepe soluții, respectând standarde relevante, pentru probleme de inginerie de complexitate medie care îndeplinesc nevoile specificate, respectând cerințe de sănătate publică, siguranță,	Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer. Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor. Studentul/absolventul comunică eficient despre activitățile de inginerie cu o gamă largă de public. Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. Studentul/absolventul promovează dialogul, cooperarea, respectul față de ceilalți și interculturalitatea. Studentul/absolventul

	bunăstare, mediu, sustenabilitate și factori economici, precum și alte constrângeri specifice. - Studentul/absolventul aplică tehnici moderne de management de proiect, tehnici economice și de luare a deciziilor inclusiv într-un cadru multidisciplinar.	lucrează eficient ca membru în echipă sau lider al acesteia.

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

Obiectivul general al disciplinei	Însușirea unor aspecte complexe ale limbajului Python și creșterea capacității de dezvoltarea sistematică a programelor

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observații
Introducere în limbajul Python		expunere orală, conversație, exemplu demonstrativ, sinteza cunoștințelor, descoperire dirijată, exemplificare, studiu de caz	
Caracteristici generale ale limbajului Python. Compilator vs. interpretor. Variabile. Tipuri de date. Operatori.	2		
Structuri de control în Python. Sintaxa generala. Particularități. Exemple. Aplicații	2		
Manipularea șirurilor de caractere în Python	2		
Structuri de date în Python. Tipuri de structuri de date (colecții). Liste în Python. Vectori si matrice manipulați prin liste. Tuple. Mulțimi. Dicționare	4		
Programare procedurală. Definirea funcțiilor in Python. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor). Aplicații.	2		
Programare modulară. Operații I/O în Python. Lucrul cu module Python. Operații de I/O cu fișiere text, binare, CSV	2		
Procesarea si vizualizarea datelor utilizând limbajul Python	2		
Instrumente Python pentru procesarea datelor			
Instrumente Python pentru vizualizarea datelor			
Folosirea instrumentelor de dezvoltare Python pentru dezvoltarea proiectelor hardware			
Instrumente Python pentru controlul echipamentelor hardware.	2		
Aplicații demonstrative.	2		
Grafuri în Python	2		
Tehnici de programare			
Metoda Greedy. Prezentare metodă. Aplicații	2		
Metoda Backtracking. Prezentare metoda. Aplicații	2		
Metoda Backtracking. Aplicații	2		

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

Aplicații (seminar / laborator / lucrări practice / proiect)	Nr. ore	Metode de predare	Observații
1. Noțiuni de sănătate și securitate în muncă. Noțiuni de prim ajutor în caz de accident. Prezentarea laboratorului. Prezentarea mediului de dezvoltare Python	2	lucrări practice, dezvoltarea de programe pe calculator, experimentul	În cadrul lucrărilor de laborator se urmărește ca programele realizate de studenţi să respecte o serie de reguli de bază ale ingineriei software. Temele de casă vor consta din dezvoltarea unor programe de complexitate medie, crearea documentaţiei aferente produsului program realizat. Etape principale în
2. Structuri de control în Python (I)	2		
3. Structuri de control în Python (II)	2		
3. Manipularea șirurilor de caractere în Python	2		
5. Aplicații folosind colecții în Python (liste, tuple)	2		
6. Test de evaluare (1)	2		
7. Dicționare în Python	2		
8. Funcții în Python	2		
9. Operații cu fișiere text.	2		
10. Utilizarea colecțiilor iterative pentru manipularea volumelor mari de date	2		
11. Utilizare module Python pentru procesarea datelor	2		

12. Aplicație metoda Greedy. Problema rucsacului.	2		dezvoltarea temei: ● însuşirea temei de realizat; ● proiectarea logică a produsului program; ● stabilirea structurilor de date şi a algoritmului; ● inserarea de comentarii în codul sursă ● punerea la punct a produsului program; testarea cu date impuse.
13. Aplicație metoda Backtracking. Problema arhipelagului de insule	2		
14. Test de evaluare (2)	2		

Prezența la activitățile obligatorii este reglementată de "R12. Regulamentul cadru privind evaluarea studenților" (https://usv.ro/calitate/pagini/regulament_usv/R12_Ed%204rev0_30.04.2020_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, 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

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Evaluare finală	Cunoașterea structurilor de date fundamentale și capacitatea de utilizare a acestora în probleme complexe.	Evaluare prin probă finală mixta (proba scrisa si proba practica pe calculator susţinută oral)	50%
Evaluare pe parcurs	Capacitatea de implementare modularizată a produselor program.	Evaluare continuă (prin metode orale şi probe practice)	50%

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
18.09.2026	s.l. dr. ing. Rusu Ionela	s.l. dr. ing. Rusu Ionela

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
22.09.2026	Ş.l.dr.ing. Corneliu BUZDUGA

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
24.09.2026	Conf.dr.ing. Eugen COCA

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
25.09.2026	Prof.dr.ing. Laurenţiu-Dan MILICI