

FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul	Departamentul de Calculatoare, Electronică și Automatică
Domeniul de studii	Inginerie electronică, telecomunicații și tehnologii informaționale
Ciclul de studii	Licență
Programul de studii	Rețele și software de telecomunicații / Electronică aplicată

2. Date despre disciplină

Denumirea disciplinei	INFORMATICĂ APLICATĂ				
Anul de studiu	I	Semestrul	2	Tipul de evaluare	E
Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categorია 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	4	Curs	2	Seminar		Laborator/lucrări practice	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	56	Curs	28	Seminar		Laborator/lucrări practice	28	Proiect	

Distribuția fondului de timp pe semestru	ore
II.a) Studiu individual	41
II.b) Tutoriat (pentru ID)	
III. Examinări	3
IV. Alte activități (precizați): Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	

Total ore studiu individual (II.a+II.b+III)	44
Total ore pe semestru (I.b+II.a+II.b+III+IV)	100
Numărul de credite	4

4. Competențe specifice acumulate

Competențe profesionale	CP 7 - Identifica nevoile clientilor CP 8 - Executa calcule matematice analitice CP 9 - Stabilește procese de date CP 21 Aplica tehnici de analiza statistica
Competențe transversale	CT 1 - Efectueaza calcule

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 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 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, bunăstare, mediu, sustenabilitate și factori economici, precum și alte constrângeri specifice.	Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate.
--	---	---

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

Obiectivul general al disciplinei	Dezvoltarea competențelor practice și teoretice necesare pentru utilizarea limbajului Python în proiectarea, implementarea și testarea aplicațiilor informatice destinate procesării datelor și controlului echipamentelor și sistemelor medicale, printr-o abordare modularizată și orientată spre integrarea software-hardware.
-----------------------------------	---

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. Caracteristici generale ale limbajului Python. Compiler vs. interpretor. Variabile. Tipuri de date. Operatori.	3		
2. Structuri de control în Python. Sintaxa generala. Particularități. Exemple. Aplicații	3		
3. Manipularea șirurilor de caractere în Python	3		
II. Structuri de date în Python.			
1. Tipuri de structuri de date (colecții). 2. Liste în Python. Vectori si matrice manipulați prin liste. 3. Tuple. Mulțimi. Dicționare	4		
III. Programare procedurală.			
1. Definirea funcțiilor in Python. 2. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor). 3. Aplicații.	3		
IV. Programare modulară.			
1. Operații I/O în Python. 2. Lucrul cu module Python. 3. Operații de I/O cu fișiere text, binare, CSV	4		
V. Procesarea si vizualizarea datelor utilizând limbajul Python			
1. Instrumente Python pentru procesarea datelor	2		
2. Instrumente Python pentru vizualizarea datelor	2		
VI. Folosirea instrumentelor de dezvoltare Python pentru dezvoltarea proiectelor hardware			
1. Instrumente Python pentru controlul echipamentelor hardware. 2. Aplicații demonstrative.	4		
Bibliografie minimală recomandată			
1. Luciano Ramalho, Fluent Python. Clear, Concise, and Effective Programming, O'Reilly Media, August 2015. 2. Python Software Foundation - https://www.python.org/ , versiune electronică, 2025 3. Tutoriale Python: http://docs.python.org/tutorial/index.html , https://www.codecademy.com/learn/learn-python , versiune electronică, 2023 4. MicroPython – Python 3 for microcontrollers - https://micropython.org/ 5. Python Resources for NI Hardware and Software - https://www.ni.com/ro-ro- support/documentation/supplemental/16/python-resources-for-ni-hardware-and-software.html 6. PyVISA: Control your instruments with Python - https://pyvisa.readthedocs.io/en/latest/			

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	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 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.
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. Control hardware utilizând limbajul Python (I)	2		
13. Control hardware utilizând limbajul Python (II)	2		
14. Test de evaluare (2)	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, 2023 5. The Python standard library. http://docs.python.org/py3k/library/index.html , versiune electronică, 2022 7. Tutoriale Python: http://docs.python.org/tutorial/index.html , https://www.codecademy.com/learn/learn-python , versiune electronică, 2022 8. John V. Guttag, Introduction to Computation and Programming Using Python. The MIT Press (2021) 9. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press (2022)			

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	- Cunoașterea elementelor de bază și a structurilor de date specifice limbajului Python; - Capacitatea de a elabora și implementa un algoritm pentru rezolvarea unei probleme - Capacitatea de utilizare a instrumentelor Python în diverse aplicații.	Evaluare prin probă finală mixta (proba scrisă și proba practică pe calculator)	50%
Laborator / lucrări practice	- Capacitatea de implementare a unui program în limbajul Python. - Capacitatea de a utiliza resurse Python pentru îndeplinirea unor cerințe impuse.	Evaluare continuă (prin metode orale și 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
22.09.2025	Ș.I. dr. ing. Ionela RUSU	Ș.I. dr. ing. Dragoș VICOVEANU

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
23.09.2025	Conf. dr. ing. Eugen COCA

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Conf. dr. ing. Eugen COCA

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