

FIȘA DISCIPLINEI (licență)

1. Date despre program

Instituția de învățământ superior	Universitatea „Ștefan cel Mare” din Suceava
Facultatea	Inginerie Electrică și Știința Calculatoarelor
Departamentul	Calculatoare, Electronică și Automatică
Domeniul de studii	Ingineria sistemelor
Ciclul de studii	Licență
Programul de studii	Automatică și informatică aplicată

2. Date despre disciplină

Denumirea disciplinei	PROGRAMAREA CALCULATOARELOR SI LIMBAJE DE PROGRAMARE II				
Titularul activităților de curs	Ș.l. dr. ing. Ionela RUSU				
Titularul activităților aplicative	As. univ. drd. ing. Mihail TARENTI				
Anul de studiu	I	Semestrul	2	Tipul de evaluare	E
Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DD - în domeniu, DS - de specialitate, DC – complementară				DF
	Categorია de opționalitate a disciplinei: DI - impusă, DO - opțională, DF - facultativă				DI

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	5	Curs	3	Seminar	Laborator/lucrări practice	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	70	Curs	42	Seminar	Laborator/lucrări practice	28	Proiect	

II Distribuția fondului de timp pe semestru:	ore
II a) Studiul după manual, suport de curs, bibliografie și notițe	14
II b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	10
II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	28
II d) Tutoriat	
III Examinări	3
IV Alte activități (precizați):	

Total ore studiu individual II (a+b+c+d)	52
Total ore pe semestru (Ib+II+III+IV)	125
Numărul de credite	5

4. Precondiții (acolo unde este cazul)

Curriculum	Programarea calculatoarelor și limbaje de programare I
Competente	C1. Utilizarea de cunoștințe de matematică, fizică, tehnica măsurării, grafică tehnică, inginerie mecanică, chimică, electrică și electronică în ingineria sistemelor. C2. Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor C5. Dezvoltarea de aplicații și implementarea algoritmilor și structurilor de conducere automata, utilizând principiile de management de proiect, medii de programare și tehnologii bazate pe microcontrolere, procesoare de semnal, automate programabile, sisteme încorporate

5. Condiții (acolo unde este cazul)

Desfășurare a cursului	PC, video proiector, exemple de programe funcționale pe calculator, suport de curs în format electronic, conexiune la internet
Desfășurare	Seminar

aplicații	Laborator/ lucrări practice	Laborator dotat cu minim 12 calculatoare având instalat mediul de dezvoltare pentru limbajul Python, ghid de lucrări practice în format electronic, conexiune la internet
	Proiect	

6. Competențe specifice acumulate

Competențe profesionale	<p>C2. Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor</p> <p>C3. Utilizarea fundamentelor automatizării, a metodelor de modelare, simulare, identificare și analiză a proceselor, a tehnicilor de proiectare asistată de calculator.</p> <p>C4. Proiectarea, implementarea, testarea, utilizarea și mentenanța sistemelor cu echipamente de uz general și dedicat, inclusiv rețele de calculatoare, pentru aplicații de automată și informatică aplicată.</p> <p>C5. Dezvoltarea de aplicații și implementarea algoritmilor și structurilor de conducere automată, utilizând principii de management de proiect, medii de programare și tehnologii bazate pe microcontrolere, procesoare de semnal, automate programabile, sisteme încorporate</p>
Competențe transversale	

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

Obiectivul general al disciplinei	Înșuirea unor aspecte complexe ale limbajului Python și creșterea capacității de dezvoltare sistematică a programelor
-----------------------------------	---

8. Conținuturi

Curs	Nr. ore	Metode de predare	Observații	
I. Introducere în limbajul Python				
1. Caracteristici generale ale limbajului Python. Compiler vs. interpretor. Variabile. Tipuri de date. Operatori.	3	expunere orală, conversație, exemplu demonstrativ, sinteza cunoștințelor, descoperire dirijată, exemplificare, studiu de caz		
2. Structuri de control în Python. Sintaxa generală. Particularități. Exemple. Aplicații	3			
3. Manipularea șirurilor de caractere în Python	3			
4. Structuri de date în Python. Tipuri de structuri de date (colecții). Liste în Python. Vectori și matrice manipulați prin liste. Tuple. Mulțimi. Dicționare	4			
5. Programare procedurală. Definirea funcțiilor în Python. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor). Aplicații.	3			
6. Programare modulară. Operații I/O în Python. Lucrul cu module Python. Operații de I/O cu fișiere text, binare, CSV	4			
II. Procesarea și vizualizarea datelor utilizând limbajul Python				
1. Instrumente Python pentru procesarea datelor				
2. Instrumente Python pentru vizualizarea datelor				
III. Folosirea instrumentelor de dezvoltare Python pentru dezvoltarea proiectelor hardware				
1. Instrumente Python pentru controlul echipamentelor hardware.	3			
2. Aplicații demonstrative.	3			
IV. Grafuri în Python				
V. Tehnici de programare				
1. Metoda Greedy. Prezentare metodă. Aplicații	3			
2. Metoda Backtracking. Prezentare metodă. Aplicații	3			
3. Metoda Backtracking. Aplicații	3			
Bibliografie				
<p>1. Luciano Ramalho, Fluent Python. Clear, Concise, and Effective Programming, O'Reilly Media, August 2015.</p> <p>2. Rance D. Necaise, Data Structures and Algorithms Using Python, Ed. JOHN WILEY & SONS, 2011.</p> <p>3. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Data Structures and Algorithms in Python, John Wiley & Sons, 2013.</p> <p>4. Python Software Foundation - https://www.python.org/, versiune electronică, 2022</p>				

5. The Python language reference. <http://docs.python.org/py3k/reference/index.html>, versiune electronică, 2022
6. 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)
10. MicroPython – Python 3 for microcontrollers - <https://micropython.org/>
11. 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>
12. PyVISA: Control your instruments with Python - <https://pyvisa.readthedocs.io/en/latest/>

Bibliografie minimală

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, 2022

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 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: <ul style="list-style-type: none"> • 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. Aplicație metoda Greedy. Problema rucsacului.	2		
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

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, 2024
5. The Python language reference. <http://docs.python.org/py3k/reference/index.html>, versiune electronică, 2024
6. The Python standard library. <http://docs.python.org/py3k/library/index.html>, versiune electronică, 2024
7. Tutoriale Python: <http://docs.python.org/tutorial/index.html>, <https://www.codecademy.com/learn/learn-python>, versiune electronică, 2024
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)

Bibliografie minimală

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, 2024

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

- Limbajul Python se află în pe primul loc în topul limbajelor de programare conform The TIOBE Programming Community index (septembrie 2022, septembrie 2023, septembrie 2024) și IEEE Spectrum (august 2023, august 2024), limbajul Python înregistrând o creștere constantă a popularității în ultimii ani.
- Cursul oferă cunoștințele de programare medii spre avansate în limbajul Python necesare pentru rezolvarea problemelor evidențiate de potențialii angajatori.
- Conținutul disciplinei asigură cunoștințele fundamentale necesare pentru utilizarea tipurilor de date abstracte în proiectarea aplicațiilor.
- *Compatibilitate națională.* Conținut similar cu cursul de Structuri de date din anul I - AIA UPB, Structuri de date și algoritmi, an 2 AIA – Universitatea Politehnică Timisoara, Programarea calculatoarelor IC UT Iasi, Tehnici de programare, anul IC, U.P. Timisoara, programarea calculatoarelor-tehnici de programare, anul IC, Universitatea Craiova.
- *Compatibilitate internațională:*
- Harvard University (SUA): CS50 - Introduction to Programming with Python;
- Massachusetts Institute of Technology (SUA): 6.006 Introduction to Algorithms, 6.100A/B Introduction to CS and Programming using Python;
- Stanford School of Engineering (SUA): CS106A - Programming Methodology, CS106B -Programming Abstractions, XCME009 - Introduction to Python, CS106X - Programming Abstractions (Accelerated).

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Cunoașterea structurilor de date fundamentale și capacitatea de utilizare a acestora în probleme complexe.	Evaluare prin probă finală mixta (proba scrisă și proba practică pe calculator susținută oral)	50%
Seminar			
Laborator/ lucrări practice	Capacitatea de implementare modularizată a produselor program.	Evaluare continuă (prin metode orale și probe practice)	50%

10.1. Standard minim de performanță evaluare la curs

- o bună descompunere a problemei în subprobleme mai simple;
- parametrizarea programelor, realizarea unei interacțiuni optime dintre date și proceduri;
- accesul la date prin funcții ce implementează operații elementare asupra acestora.
- utilizarea de instrumente software pentru procesarea datelor și dezvoltarea aplicațiilor hardware.

10.2. Standard minim de performanță evaluare la activitatea aplicativă

- rezolvarea unor probleme simple în limbajul Python;
- asigurarea de comentarii suficiente permițând o întreținere facilă a programului realizat;
- adaptarea unei concepții modulare în elaborarea programelor; reutilizarea unor module deja testate.

Data completării	Semnătura titularului de curs	Semnătura titularului de aplicație
18.09.2024		

Data avizării	Semnătura responsabilului de program
20.09.2024	

Data avizării în departament	Semnătura directorului de departament
23.09.2024	

Data aprobării în consiliul facultății	Semnătura decanului
27.09.2024	