

FIȘA DISCIPLINEI
(licență)

1. Date despre program

Instituția de învățământ superior	Universitatea „Ștefan cel Mare” Suceava
Facultatea	Inginerie Electrică și Știința Calculatoarelor
Departamentul	Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licență, învățământ cu frecvență
Programul de studii	Calculatoare

2. Date despre disciplină

Denumirea disciplinei	PROGRAMAREA CALCULATOARELOR SI LIMBAJE DE PROGRAMARE II				
Titularul activităților de curs	Ș.I. dr. ing. Ionela RUSU				
Titularul activităților aplicative	Ș.I. dr. ing. Ionela RUSU				
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	5,5	Curs	3	Seminar		Laborator/lucrări practice	2,5	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	77	Curs	42	Seminar		Laborator/lucrări practice	35	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	24
II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	32
II d) Tutoriat	
III Examinări	3
IV Alte activități (precizați):	

Total ore studiu individual II (a+b+c+d)	70
Total ore pe semestru (Ib+II+III+IV)	150
Numărul de credite	6

4. Precondiții (acolo unde este cazul)

Curriculum	Programarea calculatoarelor și limbaje de programare I
Competente	Abilități medii de programare

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 aplicații	Seminar	
	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	CP1. Operarea cu fundamente științifice, ingineresti și ale informaticii CP2. Proiectarea componentelor hardware, software și de comunicații CP3. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații
-------------------------	--

Competențe transversale	?
-------------------------	---

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

Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Însușirea unor aspecte complexe ale limbajului Python și creșterea capacității de dezvoltare sistematică a programelor • Dezvoltarea deprinderilor de proiectare structurată a programelor și aprofundarea noțiunilor de tipuri structurate de date, fișiere • Cunoașterea structurilor de date fundamentale: listă, stivă, coadă, mulțime • Însușirea metodelor Greedy și Backtracking de rezolvare a problemelor de programare • Dezvoltarea capacităților de implementare modularizată a produselor program.
-----------------------------------	---

8. Conținuturi

Curs	Nr. ore	Metode de predare	Observații	
I. Introducere în limbajul Python				
1. Tipuri și structuri de date. Variabile locale, globale, tratarea excepțiilor. Structuri de control.	3	expunere orală, conversație, exemplu demonstrativ, sinteza cunoștințelor, descoperire dirijată, exemplificare, studiu de caz		
2. Definirea funcțiilor în Python. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor). Recursivitate în Python. Aplicații	3			
3. Manipularea șirurilor de caractere în Python	2			
4. Containeri în Python	4			
5. Programare modulară. Operații I/O în Python	3			
II. Introducere în TDA (tipuri de date abstracte)				
1. Abstractizarea și încapsularea datelor. Structuri statice, structuri dinamice. Tipuri de date abstracte: domeniu, operații și reprezentarea datelor. Proiectarea TDA-urilor	3			
2. TDA Lista	2			
3. TDA stivă	2			
4. TDA coadă	2			
III. Grafuri în Python	3			
IV. Tehnici de programare				
1. Metoda Greedy. Prezentare metodă. Aplicații	3			
2. Metoda Backtracking. Prezentare metoda. Aplicații	3			
3. Metoda Backtracking. Aplicații	3			
V. Procesarea și vizualizarea datelor utilizând limbajul Python				
1. Instrumente Python pentru procesarea datelor	3			
2. Instrumente Python pentru vizualizarea datelor	3			

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, 2022
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. Mark Lutz. Learning Python. O'Reilly. 4th Edition. 2009

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. Norme de protecția muncii. Introducere în Python. Prezentarea mediului de dezvoltare Python	2.5	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	2.5		
3. Manipularea șirurilor de caractere în Python	2.5		
4. Aplicații folosind colecții în Python (liste, tuple)	2.5		
5. Manipularea funcțiilor în Python	2.5		
6. Test de evaluare (1)	2.5		
7. Dicționare în Python	2.5		
8. Operații cu fișiere text. Utilizarea colecțiilor iterative pentru manipularea volumelor mari de date	2.5		
9. Implementarea TDA listă utilizând limbajul Python	2.5		
10. Implementarea TDA stivă utilizând limbajul Python	2.5		
11. Implementarea TDA coadă utilizând limbajul Python	2.5		
12. Aplicație metoda Greedy. Problema rucsacului.	2.5		
13. Aplicație metoda Backtracking. Problema arhipelagului de insule	2.5		
14. Test de evaluare (2)	2.5		

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

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
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. Mark Lutz. Learning Python. O'Reilly. 4th Edition. 2009

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

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, 2023, 2024) și IEEE Spectrum (august 2023, 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 IC UPB, 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

10.1. Standard minim de performanță evaluare la curs

10.2. Standard minim de performanță evaluare la activitatea aplicativă

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 scrisa si proba practica pe calculator)	50%
Seminar			
Laborator / lucrări practice	Capacitatea de implementare modularizată a produselor program.	Evaluare continuă (prin metode orale și probe practice)	50%
Proiect			
Standard minim de performanță			
Standard minim de performanță evaluare la curs:			
<ul style="list-style-type: none"> • bună descompunere a problemei în subprobleme mai simple; • parametrizarea programelor, realizarea unei interacțiuni optime dintre date și proceduri; • adaptarea unei concepții modulare în elaborarea programelor • accesul la date prin funcții ce implementează operații elementare asupra acestora. • rezolvarea unor probleme complexe prin tehnicile Backtracking și Greedy în Python; 			
Standard minim de performanță evaluare la activitatea aplicativă:			
<ul style="list-style-type: none"> • rezolvarea unor probleme simple în limbajul Python; • asigurarea de comentarii suficiente permițând o întreținere facilă a programului realizat; • reutilizarea unor module deja testate. • implementarea minimală în limbajul Python a TDA listă, stivă, coadă. 			

Data completării	Semnătura titularului de curs	Semnătura cadrului didactic coordonator
23.09.2024		

Data avizării	Semnătura responsabilului de program
24.09.2024	

Data avizării în departament	Semnătura directorului de departament
25.09.2024	

Data aprobării în consiliul facultății	Semnătura decanului
27.09.2024	