

FIȘA DISCIPLINEI (licență)

1. Date despre program

Instituția de învățământ superior	Universitatea „Ștefan cel Mare” Suceava
Facultatea	Inginerie Electrică și Știința Calculatoarelor
Departamentul	Calculatoare, Electronică și Automatică
Domeniul de studii	Științe inginerești aplicate
Ciclul de studii	Licență, învățământ cu frecvență
Programul de studii	Echipamente și sisteme medicale

2. Date despre disciplină

Denumirea disciplinei	INFORMATICĂ APLICATĂ				
Titularul activităților de curs	Ș.l. dr. ing. Ionela RUSU				
Titularul activităților aplicative	Ș.l. dr. ing. Dragoș VICOVEANU				
Anul de studiu	I	Semestrul	2	Tipul de evaluare	Colocviu
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	4	Curs	2	Seminar		Laborator	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	56	Curs	28	Seminar		Laborator	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	6
II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	21
II d) Tutoriat	
III Examinări	3
IV Alte activități (precizați):	

Total ore studiu individual II (a+b+c+d)	41
Total ore pe semestru (Ib+II+III+IV)	100
Numărul de credite	4

4. Precondiții (acolo unde este cazul)

Curriculum	
Competențe	<ul style="list-style-type: none"> Abilități medii de programare

5. Condiții (acolo unde este cazul)

Desfășurare a cursului	<ul style="list-style-type: none"> PC, video proiector, exemple de programe funcționale pe calculator, suport de curs în format electronic, conexiune la internet 	
Desfășurare aplicații	Seminar	<ul style="list-style-type: none">
	Laborator	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none">

6. Competențe specifice acumulate

Competențe profesionale	C2. Utilizarea adecvată a metodelor de analiză în elaborarea și interpretarea documentației tehnologice, tehnice și inginerești
-------------------------	---

	C3. Evaluarea, punerea în funcțiune, exploatarea și întreținerea sistemelor și echipamentelor medicale din laboratoare, cabinete, clinici și spitale în condiții de securitate C5. Analiza, proiectarea tehnică și tehnologică a proceselor privind structurile și sistemele din domeniul informatic, electric, electronic și mecanic din mediul sanitar în condiții de calitate date
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 dezvoltarea sistematică a programelor • Dezvoltarea deprinderilor de proiectare structurată a programelor și aprofundarea noțiunilor de tipuri structurate de date; • Dezvoltarea capacităților de implementare modularizată a produselor program; • Dobândirea cunoștințelor în utilizarea de instrumente software pentru procesarea datelor și dezvoltarea aplicațiilor hardware.
-----------------------------------	--

8. Conținuturi

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. Compilator vs. interpretor. Variabile. Tipuri de date. Operatori.	3		
2. Structuri de control în Python. Sintaxa generală. 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).	4		
2. Liste în Python. Vectori și matrice manipulați prin liste.			
3. Tuple. Mulțimi. Dicționare			
III. Programare procedurală.			
1. Definirea funcțiilor în Python.	3		
2. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor).			
3. Aplicații.			
IV. Programare modulară.			
1. Operații I/O în Python.	4		
2. Lucrul cu module Python.			
3. Operații de I/O cu fișiere text, binare, CSV			
V. Procesarea și 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.	4		
2. Aplicații demonstrative.			
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

11. MicroPython – Python 3 for microcontrollers - <https://micropython.org/>

12. 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>

13. 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 / 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:
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		

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, 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) și IEEE Spectrum (septembrie 2022), 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 este în concordanță cu ceea ce se predă în cadrul unor facultăți de profil electric din alte centre universitare din țară și din străinătate. Conținut similar cu cursul de Structuri de date din anul I - AIA UPB, Structuri de date și algoritmi, an 2 AIA – Universitatea Politehnica Timisoara, Programarea calculatoarelor și limbaje de programare II – anul I Inginerie Medicală, Universitatea Tehnică Cluj, 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	<ul style="list-style-type: none"> - 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 scrisă	50%
Seminar			
Laborator	<ul style="list-style-type: none"> - 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%
Proiect			
Standard minim de performanță			
<ul style="list-style-type: none"> • rezolvarea unor probleme simple în limbajul Python; • o bună descompunere a problemei în subprobleme mai simple; • asigurarea de comentarii suficiente permițând o întreținere facilă a programului realizat. • adaptarea unei concepții modulare în elaborarea programelor; reutilizarea unor module deja testate; • utilizarea de instrumente Python în dezvoltarea de aplicații. 			

Data completării	Semnătura titularului de curs	Semnătura titularului de aplicație
15.09.2022	Ș.l. dr. ing. Ionela RUSU	Ș.l. dr. ing. Dragoș VICOVEANU

Data avizării în departament	Semnătura directorului de departament
26.09.2022	Conf.univ.dr.ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Semnătura decanului
30.09.2022	Prof.univ.dr.ing. Laurențiu Dan MILICI

