

## 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
Programul de studii	Program de conversie profesională – Tehnologia informației și comunicațiilor / Profesor TIC

### 2. Date despre disciplină

Denumirea disciplinei	STRUCTURI DE DATE				
Titularul activităților de curs	Ș.I. dr. ing. Ionela RUSU				
Titularul activităților aplicative	Drd. Mihail TERENTI				
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				DD
	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/lucrări practice	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	56	Curs	28	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	63
II b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	36
II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	42
II d) Tutoriat	
III Examinări	3
IV Alte activități (precizați):	

Total ore studiu individual II (a+b+c+d)	141
Total ore pe semestru (Ib+II+III+IV)	200
Numărul de credite	8

### 4. Precondiții (acolo unde este cazul)

Curriculum	• Nu este cazul
Competente	• Nu este cazul

### 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

Competente	C1. Operarea cu fundamente științifice și ale informaticii
------------	--

profesionale	C2. Proiectarea componentelor software C4. Îmbunătățirea performanțelor sistemelor informatice
Competențe transversale	CT3. Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale și de cultură organizațională

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

Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Însușirea unor aspecte complexe ale limbajului Python și creșterea capacității de dezvoltarea sistematică a programelor</li> </ul>
	<ul style="list-style-type: none"> <li>• Dezvoltarea deprinderilor de proiectare structurată a programelor și aprofundarea noțiunilor de tipuri structurate de date;</li> <li>• Dezvoltarea capacităților de implementare modularizată a produselor program;</li> </ul>

### 8. Conținuturi

Curs	Nr. ore	Metode de predare	Observații	
<b>I. Introducere în limbajul Python</b>				
1. Caracteristici generale ale limbajului Python. Compilator 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			
<b>II. Structuri de date în Python.</b>				
1. Tipuri de structuri de date (colecții).	4			
2. Liste în Python. Vectori și matrice manipulați prin liste.				
3. Tuple. Mulțimi. Dicționare				
<b>III. Programare procedurală.</b>				
1. Definierea funcțiilor în Python.	3			
2. Funcții cu număr variabil de argumente (împachetarea și despachetarea argumentelor).				
3. Aplicații.				
<b>IV. Programare modulară.</b>				
1. Operații I/O în Python.	4			
2. Lucrul cu module Python.				
3. Operații de I/O cu fișiere text, binare, CSV				
<b>V. Procesarea și vizualizarea datelor utilizând limbajul Python</b>				
1. Instrumente Python pentru procesarea datelor	2			
2. Instrumente Python pentru vizualizarea datelor	2			
<b>VI. Grafuri în Python</b>				
1. Teorie și aplicații demonstrative	4			

#### 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. 4<sup>th</sup> 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, 2023

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: <ul style="list-style-type: none"> <li>• însușirea temei de realizat;</li> <li>• proiectarea logică a produsului program;</li> <li>• stabilirea structurilor de date și a algoritmului;</li> <li>• inserarea de comentarii în codul sursă</li> <li>• punerea la punct a produsului program;</li> <li>• testarea cu date impuse.</li> </ul>
2. Structuri de control în Python (I)	2		
3. Structuri de control în Python (II)	2		
4. Manipularea șirurilor de caractere în Python	2		
5. Aplicații folosind colecții în Python (liste, tuple)	2		
6. Implementare TDA lista	2		
7. Implementare TDA Stiva	2		
8. Test de evaluare (1)	2		
9. Dicționare în Python	2		
10. Funcții în Python	2		
11. Operații cu fișiere text.	2		
12. Utilizarea colecțiilor iterative pentru manipularea volumelor mari de date	2		
13. Grafuri. Aplicații	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](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, 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)

**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, 2023

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

- Cursul oferă cunoștințele de programare și utilizare a instrumentelor oferite de limbajul Python necesare pentru viitorii absolvenți în procesul de predare a disciplinelor cu specific TIC.
- Cursul este compatibil cu alte cursuri din țară din cadrul programelor de formare profesională cu specific TIC

**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	<ul style="list-style-type: none"> <li>- Cunoașterea elementelor de bază și a structurilor de date specifice limbajului Python;</li> <li>- capacitatea de a elabora și implementa un algoritm pentru rezolvarea unei probleme</li> <li>- capacitatea de utilizare a instrumentelor Python în diverse aplicații.</li> </ul>	Evaluare prin probă finală mixta (proba scrisă și proba practică pe calculator)	50%
Seminar			
Laborator / lucrări practice	<ul style="list-style-type: none"> <li>- Capacitatea de implementare a unui program în limbajul Python.</li> <li>- Capacitatea de a utiliza resurse Python pentru îndeplinirea unor cerințe impuse.</li> </ul>	Evaluare continuă (prin metode orale și probe practice)	50%
Proiect			
<b>Standard minim de performanță</b>			
Standard minim de performanță evaluare la curs			
<ul style="list-style-type: none"> <li>- rezolvarea unor probleme simple în limbajul Python;</li> <li>- o bună descompunere a problemei în subprobleme mai simple;</li> <li>- asigurarea de comentarii suficiente permițând o întreținere facilă a programului realizat.</li> </ul>			
Standard minim de performanță evaluare la activitatea aplicativă:			
<ul style="list-style-type: none"> <li>- adaptarea unei concepții modulare în elaborarea programelor;</li> <li>- reutilizarea unor module deja testate;</li> <li>- utilizarea de instrumente Python în dezvoltarea de aplicații.</li> </ul>			

Data completării	Semnătura titularului de curs	Semnătura cadrului didactic coordonator
<b>23.09.2024</b>		

Data avizării	Semnătura responsabilului de program
<b>24.09.2024</b>	

Data avizării în departament	Semnătura directorului de departament
<b>25.09.2024</b>	

Data aprobării în consiliul facultății	Semnătura decanului
<b>27.09.2024</b>	