

	CURRICULUMUL DISCIPLINEI	Cod:	CD - 8.1
	DENUMIREA DISCIPLINEI	Pagina	1 din 10

**FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ
DEPARTAMENTUL INGINERIA SOFTWARE ȘI AUTOMATICĂ**

APROBAT

la ședința Departamentului ISA

nr. 1 din 26.08.2025

Șef departament

FIODOROV Ion

conf. univ., dr.

APROBAT

la ședința Consiliului Facultății CIM

nr. 1 din 26.08.2025

Președintele Consiliului CIM

CIORBĂ Dumitru,

conf. univ., dr.

Program de studii: *0613.1 Tehnologia informației*

Cod, Denumirea modulului: *S.O.001 Tehnici de programare*

Beneficiari: *Studentii anului I, învățământ cu frecvență și învățământ cu frecvență redusă*

Ciclul de învățământ: *Studii superioare de licență, ciclul I*

Numărul de credite ECTS: *5*

Titularul disciplinei

lect. univ., dr. MORARU Dumitru / _____
Nume, prenume, semnătura titularului

I. PRELIMINARII

Studiul tehnicilor de programare este un element de bază pentru formarea competențelor de început în domeniul tehnologiei informației și al ingineriei moderne. Programarea reprezintă limbajul universal prin care omul comunică cu calculatorul și dezvoltă soluții pentru probleme reale, de la aplicații educaționale și sisteme de gestiune până la inteligență artificială.

Cursul oferă studenților capacitatea de a: înțelege logica și structura algoritmilor; transpune soluții teoretice în programe funcționale; dezvolta gândirea logică și analitică necesară pentru rezolvarea problemelor; utiliza instrumente software moderne pentru implementarea aplicațiilor practice.

În plus, alegerea limbajului Python ca suport didactic se bazează pe accesibilitatea sa, lizibilitatea codului și aplicabilitatea în multiple domenii actuale, de la știința datelor și inteligența artificială până la dezvoltarea web și pachete de programe complexe.

II. PRECONDIȚII DE ACCES LA DISCIPLINĂ/MODUL

Pentru a atinge obiectivele cursului, studenții trebuie să posede cunoștințe fundamentale de matematică (algebră liniară, analiză matematică), logică și geometriei analitice, precum și noțiuni introductive de calculatoare. De asemenea, este necesară familiarizarea cu conceptele de bază din informatică și programare, care permit dezvoltarea gândirii algoritmice și a capacității de prelucrare a datelor.

Accesul la curs presupune promovarea disciplinelor anterioare din planul de învățământ, cum sunt: Algebra liniară și geometria analitică, Analiza matematică 1, Programarea calculatoarelor, acestea constituind baza teoretică și practică necesară pentru însușirea tehnicilor de programare și aplicarea lor în domeniul tehnologiei informației.

III. REZULTATE ALE ÎNVĂȚĂRII CARE URMEAZĂ A FI DEZVOLTATE

Competențe Generale/Profesionale	Rezultate ale învățării conform nivelului CNC <i>Absolventul/candidatul la atribuirea calificării poate:</i>
CG 1. Utilizarea în activitatea profesională a conceptelor, teoriilor și metodelor științelor fundamentale	1. aplica conceptele și metodele științelor fundamentale pentru identificarea, formularea și argumentarea soluțiilor în domeniul ingineriei;
CG 2. Operarea cu concepte de bază din știința calculatoarelor, tehnologia informației și comunicațiilor	2. utiliza concepte matematice, statistice și logice pentru analiza și interpretarea datelor colectate în proiectele de dezvoltare a produselor program și a aplicațiilor
	3. aplica concepte de bază din știința calculatoarelor, tehnologia informației și comunicațiilor în contextul dezvoltării produselor program și a aplicațiilor
CP 1. Definirea specificațiilor sistemului informatic	4. analiza cerințele utilizatorilor și specifica cerințele funcționale și nefuncționale ale unui sistem informatic
CP 2. Implementarea sistemelor informatice	5. implementa și testa funcționalitățile unui sistem informatic

IV. ADMINISTRAREA DISCIPLINEI/MODULULUI

Cod	Anul	Semestrul	Numărul de ore					
			Curs	Seminar	Lucrări practice	Lucrări de laborator	Lucrul individual	Proiectare
S.O.001	I	învățământ cu frecvență						
		1	30		30		90	
	I	învățământ cu frecvență redusă						
		2	12		18		120	

V. REZULTATELE ÎNVĂȚĂRII, CONȚINUTURI ȘI METODE DIDACTICE APLICATE

Rezultatele învățării	Conținuturi		Metode de predare, evaluare, control	Realizarea în timp (ore)					
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență			învățământ cu frecvență redusă		
				curs	seminare	practice	curs	seminare	practice
Cunoștințe: CG2. Rî.3. K8. Limbaje, medii și tehnologii de programare; CG2. Rî.3. K9. Structuri de date și algoritmi. CP1. Rî.9. K1. Metode de identificare a cerințelor; CP1. Rî.9. K2. Principiile formulării cerințelor funcționale și nefuncționale. Aptitudini: CG2. Rî.3. S3. Clasifică și selectează arhitecturile sistemelor de calcul. CP1. Rî.9. S1. Identifică cerințele utilizatorilor folosind metode structurate.	T.1. Introducere în Tehnici de programare (Python)	L.P.1. Introducere în Tehnici de programare (Python)	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică: Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	2		2	2		2
Cunoștințe: CG2. Rî.3. K8. Limbaje, medii și tehnologii de programare; CG2. Rî.3. K9. Structuri de date și algoritmi. CP1. Rî.9. K3. Structura și elementele unui document de specificații pentru sistemul informatic. Aptitudini: CG2. Rî.3. S4. Identifică și analizează componentele sistemelor de operare.	T.2. Bazele limbajului Python și structuri de control	L.P.2. Bazele limbajului Python și structuri de control	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică: Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	2		4	2		2
Cunoștințe: CG2. Rî.3. K8. Limbaje, medii și tehnologii de programare; CG2. Rî.3. K9. Structuri de date și algoritmi.	T.3. Funcții și modularizare	L.P.3. Funcții și modularizare	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică:	2		4	2		2

Rezultatele învățării	Conținuturi		Metode de predare, evaluare, control	Realizarea în timp (ore)					
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență			învățământ cu frecvență redusă		
				curs	seminare	practice	curs	seminare	practice
<i>CP1. Rî.9. K4. Principiile de structurare a cerințelor în funcție de impactul asupra sistemului.</i> Aptitudini: <i>CG1. Rî.2. S3. Utilizează relații și operații ale algebrei relaționale;</i> <i>CG2. Rî.3. S4. Identifică și analizează componentele sistemelor de operare.</i> <i>CP1. Rî.9. S2. Analizează cerințele și le structurează în funcție de obiectivele sistemului.</i>			Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.						
Cunoștințe: <i>CG1. Rî.2. K3. Relații și operații ale algebrei relaționale;</i> <i>CG1. Rî.2. K5. Principii de vizualizare a datelor.</i> <i>CG2. Rî.3. K9. Structuri de date și algoritmi;</i> <i>CP2. Rî.12. K3. Principiile optimizării performanței unui sistem.</i> Aptitudini: <i>CG1. Rî.2. S3. Utilizează relații și operații ale algebrei relaționale;</i> <i>CG1. Rî.2. S4. Analizează structuri de date complexe utilizând teoria grafurilor;</i> <i>CG1. Rî.2. S5. Utilizează principii de vizualizare a datelor.</i> <i>CP1. Rî.9. S4. Documentează cerințele utilizând formate standardizate (fișiere, date stocate).</i>	<i>T.4. Structuri de date. Lucrul cu fișierele.</i>	<i>L.P.4. Structuri de date. Lucrul cu fișierele.</i>	<i>Pentru prelegere:</i> Expunerea, conversația. <i>Pentru lucrare practică:</i> Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	4		4	2		2

Rezultatele învățării	Conținuturi		Metode de predare, evaluare, control	Realizarea în timp (ore)					
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență			învățământ cu frecvență redusă		
				curs	seminare	practice	curs	seminare	practice
Cunoștințe: CG2. Rî.3. K8. Limbaje, medii și tehnologii de programare; CG2. Rî.3. K9. Structuri de date și algoritmi. CP1. Rî.9. K6. Conceptele de bază ale constrângerilor tehnice în dezvoltarea unui sistem informatic. Aptitudini: CG1. Rî.2. S5. Utilizează principii de vizualizare a datelor (pentru analiza logurilor); CG2. Rî.3. S4. Identifică și analizează componentele sistemelor de operare; CP1. Rî.9. S6. Utilizează metode adecvate pentru analiza constrângerilor tehnice.	T.5. Mecanisme de tratare a excepțiilor lucrul cu jurnalele	L.P.5. Mecanisme de tratare a excepțiilor lucrul cu jurnalele	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică: Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	4		4	2		2
Cunoștințe: CG2. Rî.3. K8. Limbaje, medii și tehnologii de programare; CG2. Rî.3. K9. Structuri de date și algoritmi; CP2. Rî.12. K1. Metode de implementare a funcționalităților unui sistem informatic. Aptitudini: CG2. Rî.3. S3. Clasifică și selectează arhitecturile sistemelor de calcul; CG2. Rî.3. S4. Identifică și analizează componentele sistemelor de operare; CP2. Rî.12. S1. Implementează funcționalitățile unui sistem conform cerințelor.	T.6. Elemente de programare orientată pe obiecte	L.P.6. Elemente de programare orientată pe obiecte	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică: Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	6		4	4		4

Rezultatele învățării	Conținuturi		Metode de predare, evaluare, control	Realizarea în timp (ore)					
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență			învățământ cu frecvență redusă		
				curs	seminare	practice	curs	seminare	practice
Cunoștințe: CG2. Rî.3. K8. Limbaje, medii și tehnologii de programare; CG2. Rî.3. K9. Structuri de date și algoritmi; CG1. Rî.2. K5. Principii de vizualizare a datelor. CP2. Rî.12. K1. Metode de implementare a funcționalităților unui sistem informatic. Aptitudini: CG1. Rî.2. S5. Utilizează principii de vizualizare a datelor și interpretare grafică a rezultatelor; CG2. Rî.3. S6. Aplică tehnici de analiză și prelucrare a semnalelor (relevant pentru biblioteci de procesare numerică și științifică). CP2. Rî.12. S1. Implementează funcționalitățile unui sistem conform cerințelor.	T.7. Biblioteci standard și populare	L.P.7. Biblioteci standard și populare	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică: Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	4		4	2		2
Cunoștințe: CG1. Rî.2. S5. Utilizează principii de vizualizare a datelor și interpretare grafică a rezultatelor; CG2. Rî.3. S6. Aplică tehnici de analiză și prelucrare a semnalelor (relevant pentru biblioteci de procesare numerică și științifică). CP2. Rî.12. K2. Metode și tehnici de testare a componentelor sistemelor informatice. Aptitudini:	T.8. Paradigme moderne și tendințe în programare	L.P.8. Paradigme moderne și tendințe în programare	Pentru prelegere: Expunerea, conversația. Pentru lucrare practică: Problematizarea; tehnici de învățare prin simulare pe calculator și interactiv-creativă.	6		4	4		2

Rezultatele învățării	Conținuturi		Metode de predare, evaluare, control	Realizarea în timp (ore)					
	Curs	Seminare/lucrări practice/lucrări de laborator		învățământ cu frecvență			învățământ cu frecvență redusă		
				curs	seminare	practice	curs	seminare	practice
CG1. Rî.2. S2. Aplică metode de modelare probabilistică și inferență statistică în luarea deciziilor; S4. Analizează structuri de date complexe; CG1. Rî.2. S5. Utilizează principii de vizualizare a datelor. CP2. Rî.12. S2. Testează componentele sistemului pentru a verifica funcționarea corectă									
Total				30		30	12		18

VI. SUGESTII PENTRU ACTIVITATEA INDIVIDUALĂ A STUDENȚILOR

Nr. crt.	Capitol, temă	Conținut activitate individuală	Durata, ore		Forma de control	Termeni de control (perioada)
			frecvența la zi	frecvența redusă		
1	T.1. Introducere în Tehnici de programare	Documentarea despre istoricul limbajului Python și domeniile sale de aplicare.	6	10	Discuții, prezentare raportului	Pe parcursul semestrului de studiu
2	T.2. Bazele limbajului Python și structuri de control	Rezolvarea de exerciții simple folosind tipuri de date fundamentale și instrucțiuni <i>if</i> , <i>for</i> , <i>while</i> . Cel puțin 5 exerciții pentru fiecare instrucțiune.	10	10	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
3	T.3. Funcții și modularizare	Crearea a minim 6 funcții proprii pentru calcule matematice și împărțirea codului în module separate.	10	10	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
4	T.4. Structuri de date. Lucrul cu fișierele.	Implementarea a minim 5 programe care folosesc liste și dicționare pentru stocarea datelor și salvarea rezultatelor într-un fișier text.	14	20	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
5	T.5. Mecanisme de tratare a excepțiilor lucrul cu jurnalele	Exersarea scrierii de programe care tratează erorile de intrare a utilizatorului și care generează fișiere log. Cel puțin un program care tratează minim 3 excepții	10	15	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
6	T.6. Elemente de programare orientată pe obiecte	Elaborarea a minim 3 clase simple care descriu obiecte din viața reală și testarea metodelor acestora.	14	20	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
7	T.7. Biblioteci standard și populare	Explorarea documentației pentru bibliotecile <i>math</i> , <i>datetime</i> , <i>numpy</i> , <i>pandas</i> și rezolvarea unor sarcini practice cu acestea. Cel puțin 6 programe care utilizează biblioteci integrate.	12	15	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
8	T.8. Paradigme moderne și tendințe în programare	Studierea noțiunilor de programare funcțională și scrierea unor programe scurte cu <i>lambda</i> , <i>map</i> , <i>filter</i> , <i>reduce</i> . Câte un program pentru fiecare funcție.	14	20	Discuții, prezentarea execuției programului	Pe parcursul semestrului de studiu
Total			90	120		

VII. UTILIZAREA INTELIGENȚEI ARTIFICIALE GENERATIVĂ

Permisuni de utilizare	<i>Utilizarea IA generative în cadrul temelor și proiectelor este permisă, cu condiția ca studenții să respecte următoarele reguli:</i> <i>IA generativă poate fi utilizată pentru generarea de idei, structuri de text sau cod, dar toate materialele generate trebuie să fie revizuite și ajustate de către student pentru a se asigura că acestea corespund cerințelor academice.</i> <i>Orice utilizare a IA generative trebuie să fie declarată în secțiunea de apendice a fiecărei lucrări, folosind fraza: "În timpul pregătirii acestei lucrări, autorul a utilizat [NUME INSTRUMENT / SERVICIU] în scopul [MOTIV]. După utilizarea instrument/serviciu respectiv, autorul a revizuit și editat conținutul după cum a fost necesar și își asumă întreaga responsabilitate pentru conținutul lucrării."</i>
-------------------------------	---

Restricții de utilizare	<i>Studentii nu trebuie să considere IA generativă ca o sursă de încredere pentru informații, deoarece nu oferă referințe clare sau surse documentate. Nu este permisă citarea directă a conținutului generat de IA în lucrările academice ca și cum ar fi sursă primară.</i> <i>Activitățile în care este interzis utilizarea IA generativă sunt specificare de profesor și sunt de regulă evaluări intermediare și finale sau care nu presupun activități de dezvoltare a competențelor profesionale.</i>
--------------------------------	---

VIII. CRITERII DE EVALUARE

Tip de evaluare	Modul de desfășurare, standard minim de performanță	Pondere pe componente de conținut	Nota generală
învățământ cu frecvență			
Evaluare periodică			Nota semestrială 60%
EP 1	<i>Test de tip ITAM (itemi cu alegere multiplă) în format electronic pe platforma ELSE (Moodle).</i>	15 %	
EP 2	<i>Se notează în baza unei medii aritmetice a notelor obținute la lucrări practice.</i>	15 %	
Evaluare curentă	<i>În baza prezenței la ore. Se face raportul dintre numărul maxim de ore de contact direct la disciplină și scara de notare de la 0 la 10.</i>	15 %	
Studiu individual	<i>Prezentarea tuturor rapoartelor/dovezilor de îndeplinire a sarcinilor pentru teme din capitolul VI al acestei curricule</i>	15 %	
Examen semestrial	<i>Test de tip ITAM (itemi cu alegere multiplă) în format electronic pe platforma ELSE (Moodle).</i>	40 %	Evaluare finală 40%
învățământ cu frecvență redusă			
Evaluare periodică			Nota semestrială 50%
EP 1	<i>Test de tip ITAM (itemi cu alegere multiplă) în format electronic pe platforma ELSE (Moodle).</i>	15 %	
EP 2	<i>Se notează în baza unei medii aritmetice a notelor obținute la lucrări practice.</i>	15 %	
Evaluare curentă	<i>În baza prezenței la ore. Se face raportul dintre numărul maxim de ore de contact direct la disciplină și scara de notare de la 0 la 10.</i>	10 %	
Studiu individual	<i>Prezentarea tuturor rapoartelor/dovezilor de îndeplinire a sarcinilor pentru teme din capitolul VI al acestei curricule</i>	10 %	
Examen semestrial	<i>Test de tip ITAM (itemi cu alegere multiplă) în format electronic pe platforma ELSE (Moodle).</i>	50 %	Evaluare finală 50%

IX. LISTA DE SUBIECTE PENTRU EVALUĂRI

Chestionar pentru EP 1

Pentru evaluare periodică 1 se elaborează un test de tip ITAM (itemi cu alegere multiplă) format din 50 de întrebări care reflectă materialul studiat după cum urmează:

- 1 Introducere în Tehnici de programare (Python)
- 2 Bazele limbajului Python și structuri de control
- 3 Funcții și modularizare
- 4 Structuri de date. Lucrul cu fișierele.
- 5 Mecanisme de tratare a excepțiilor lucrul cu jurnalele

Chestionar pentru EP 2

Pentru evaluare periodică 2 se va face media aritmetică a notelor obținute la lucrările practice după cum urmează temele lucrărilor:

- 1 Introducere în Tehnici de programare (Python)
- 2 Bazele limbajului Python și structuri de control
- 3 Funcții și modularizare
- 4 Structuri de date. Lucrul cu fișierele.
- 5 Mecanisme de tratare a excepțiilor lucrul cu jurnalele
- 6 Elemente de programare orientată pe obiecte
- 7 Biblioteci standard și populare

Chestionar pentru Examen semestrial

Pentru examenul semestrial se va elabora un test de tip ITAM cu un set de 100 de întrebări care vor reflecta conținutul tuturor temelor studiate, după cum urmează:

- 1 Introducere în Tehnici de programare (Python)
- 2 Bazele limbajului Python și structuri de control
- 3 Funcții și modularizare
- 4 Structuri de date. Lucrul cu fișierele.
- 5 Mecanisme de tratare a excepțiilor lucrul cu jurnalele
- 6 Elemente de programare orientată pe obiecte
- 7 Biblioteci standard și populare
- 8 Paradigme moderne și tendințe în programare

X. REFERINȚE BIBLIOGRAFICE

Obligatorii

- 1 Cotelea V., Ungur C. (2020). Python: prima mea carte, Universitatea Tehnica a Moldovei. Chisinau: S. n., Tipogr. "Foxtrot". ISBN 978-9-97-589161-5.
- 2 Cujba R., Borozan O. (2025). Bazele programării calculatoarelor în limbajul Python. ISBN 978-9975-64-557-7.
- 3 Amos D., Bader D., Jablonski J., Heisler F. (2020). Python Basics: A Practical Introduction to Python 3. 4th Edition, Real Python. ISBN 978-1-77-509333-6.

Suplimentare

- 4 Learn Python – Free Interactive Python Tutorial -<https://www.learnpython.org/>.
- 5 Bader D. (2018). Python Tricks: The Book. ISBN 978-1-77-509330-5.
- 6 Barry P. (2017). Head First Python. A Brain-Friendly Guide. O'Reilly. ISBN 978-1-491-91953-8.
- 7 Beazley D., Jones B.K. (2013). Python Cookbook. Third Edition, O'Reilly Media. ISBN 978-1-44-934037-7.
- 8 Das U., Lawson A., Mayfield C., Norouzi N. (2024). Introduction to Python Programming. Open Stax, Rice University. ISBN 978-1-96-158445-7.
- 9 Hunt J. (2019). A Beginners Guide to Python 3 Programming. Springer. ISBN 978-3-03-020290-3.
- 10 Lutz M. (2009). Learning Python, Fourth Edition. O'Reilly Media. ISBN 978-0-59-615806-4
- 11 Matthes E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Second Edition, No Starch Press. ISBN 978-1-59-327928-8.