FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea[1] / Departamentul[2]	Inginerie din Hunedoara/Inginerie Electrică și Informatică Industrială
1.3 Domeniul de studii (denumire/cod [3])	Inginerie electrică / 90
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Inginerie electrică și calculatoare / 60 / Inginer

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă[4]			Programarea calculatoarelor și limbaje de programare 1 / DF				
2.1b Denumirea disciplinei în limba engleză			Computer programming and programming languages 1				
2.2 Titularul activităţilor de curs			Șef lucr. Dr. Ing. Ghiormez Loredana				
2.3 Titularul activităţilor aplicative[5]			Șef lucr. Dr. Ing. Ghiormez Loredana				
2.4 Anul de studii[6]	I	**2.5** Semestrul	1	**2.6** Tipul de evaluare	E	**2.7** Regimul disciplinei[7]	DOB

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate) [8]

3.1 Număr de ore asistate integral/săptămână	5 , format din:	**3.2** ore curs	2	**3.3** ore seminar/laborator/proiect	3
3.1* Număr total de ore asistate integral/sem.	70 , format din:	**3.2*** ore curs	28	**3.3*** ore seminar/laborator/proiect	42
3.4 Număr de ore asistate parțial/săptămână	, format din:	**3.5** ore practică		**3.6** ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/ semestru	, format din:	**3.5*** ore practică		**3.6*** ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/ săptămână	3.93 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			0.93
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1.5
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1.5
3.7* Număr total de ore activități neasistate/ semestru	55 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			13
		ore studiu individual după manual, suport de curs, bibliografie și notițe			21
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			21
3.8 Total ore/săptămână [9]	8.93				
3.8* Total ore/semestru	125				
3.9 Număr de credite	5				

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Cunoștințe de bază de matematică.
4.2 de rezultatele învățării	• Cunoștințe generale de operare pe calculator: gestionarea fișierelor și directoarelor, instalarea aplicațiilor, utilizarea unui browser web; • Abilitatea de structurare logică a pașilor necesari rezolvării unei probleme simple din viața reală sau din matematică.

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Cursul se va desfășura într-o sală dotată cu videoproiector, tablă pentru scris și computer cu conexiune la internet.
5.2 de desfăşurare a activităţilor practice	• Sală de laborator echipată cu cu videoproiector, tablă pentru scris și computere cu conexiune la internet.

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• **C1. Studentul/absolventul identifică și descrie concepte, principii și metode de bază din matematică, fizică, economie și informatică.** C1.1. Studentul demonstrează însușirea bazelor informaticii prin reprezentarea unor algoritmi fundamentali de prelucrare numerică folosind scheme logice, limbajul pseudocod, respectiv transpunerea acestora în cod sursă scris în Python. • **C8. Studentul/absolventul identifică și descrie principiile utilizate în programare și modelare, simulare, baze de date, inteligență artificială, programare în timp real, învățare automată și rețele neuronale.** C8.1. Studentul proiectează o schemă logică, redactează pseudocodul corespunzător și implementează un script Python procedural (fără concepte de programare orientată pe obiecte) care simulează comportamentul unui sistem fizic, utilizând funcții, structuri repetitive și stocarea datelor simulate în fișiere text sau fișiere binare. •
Abilități	• **A1.Studentul/absolventul ajustează proiectele de produse sau de părți de produse astfel încât acestea să îndeplinească cerințele. Studentul/absolventul creează și/sau execută un plan sau specificație pentru proiectarea unor sisteme industriale, materiale, produse sau un plan de producție, bazate pe concepte de design estetic și/sau funcțional.** A1.1. Implementarea de algoritmi care au la baza rezolvarea unor probleme din domeniul ingineriei electrice și calculatoare, asigurând funcționarea acestora conform cerințelor specifice. • **A2. Studentul/absolventul specifică proprietăți tehnice ale bunurilor, materialelor, metodelor, proceselor, serviciilor, sistemelor, software-ului și funcționalităților, prin identificarea și răspunsul la nevoile particulare care urmează să fie satisfăcute în funcție de cerințele clienților.** A2.1. Studentul transpune cerințele unor probleme din domeniul ingineriei electrice și calculatoare folosind scheme logice, limbajul pseudocod și limbajul Pyhton. • **A10. Studentul/absolventul selectează și aplică metodele actuale de modelare, calcul, proiectare și testare pentru specializarea lor**. • A10.1. Studentul proiectează și testează algoritmi care modelează comportamentul unui sistem electric astfel încât să se asigure corectitudinea calculelor matematice și conformitatea cu specificațiile de proiectare.
Responsabilitate și autonomie	• **RA1. Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer.** RA1.1. Scrierea unui cod sursă original, sigur și robust prin gestionarea corectă a memoriei și tratarea excepțiilor, prevenind erorile critice în simularea sistemelor inginerești. • **RA2. Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor.** RA2.1. Identificarea decalajelor individuale în înțelegerea algoritmilor sau sintaxei Python și remedierea autonomă a acestora prin parcurgerea de tutoriale, resurse interactive și exerciții suplimentare. •

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

- Dezvoltarea gândirii structurate și algoritmice a studenților prin stăpânirea metodologiilor de modelare grafică (scheme logice, limbaj pseudocod) și prin implementarea acestora în limbajul de programare Python, în scopul rezolvării problemelor specifice ingineriei electrice în acord cu cerințele tehnice ale pieței muncii, normele etice și rigorile de siguranță profesională.
-

8. Conţinuturi[10]

8.1 Curs	Număr de ore	Metode de predare[11]
1. Teoria algoritmilor 1.1. Definiţie. Proprietăţi 1.2. Expresii 1.3. Operațiile de bază realizate de un algoritm 1.4. Reprezentarea algoritmilor 1.4.1. Scheme logice 1.4.2. Principiile de bază ale programării structurate 1.4.3. Limbajul pseudocod	4	Studenții au acces la resurse în format electronic, accesând campusul virtual al UPT sau folosind rețeaua locală de intranet a facultății. Se vor utiliza atât prezentări interactive cât și tradiționale.
1.5. Subalgoritmi 1.6. Recursivitate	6	
1.7. Analiza algoritmilor 1.7.1. Verificarea corectitudinii	2	

1.7.2. Calculul efectuat prin execuția unui algoritm 1.7.3. Analiza eficienței (complexității) algoritmilor		Se vor folosi: problematizarea, studiu de caz, conversația.
2. Limbajul de programare Python 2.1. Evoluția limbajelor de programare 2.2. Scurt istoric al limbajului Python 2.2.1. Cuvinte cheie 2.3. Tipuri de date și variabile 2.4. Operatori 2.4.1. Conversia numerelor întregi 2.4.2. Exemple operatori logici pe biți 2.5. Comentarii 2.6. Funcții de intrare/ieșire 2.7. Structura alternativă – instrucțiunea if 2.8. Utilizări ale operatorilor logici pe biți	2	
2.9. Instrucțiunile corespunzătoare structurii repetitive 2.10. Module Python și includerea acestora într-un program	2	
2.11. Funcții. Tipuri de variabile	2	
2.12. Colecții de date: tipuri secvențiale, mulțimi și dicționare	6	
2.13. Prelucrarea fișierelor 2.13.1. Fișiere text: csv 2.13.2. Fișiere binare.	4	

Bibliografie[12]

1. Ghiormez L., Curs în format electronic – Programarea calculatoarelor și limbaje de programare 1, 2026, Campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=3546 sau intranet \\fs\cursuri\PCLP 1 2026
2. Manuela Pănoiu, Ionel Muscalagiu, Caius Pănoiu, Utilizarea şi programarea calculatoarelor, Editura Mirton, 2002, Timişoara
3. Cristian Giumale, Introducere în analiza algoritmilor, Editura Polirom, 2004, Iaşi
4. Rod Stephens, Essential Algorithms, John Wiley & Sons Ltd, 2013
5. Magnus Lie Hetland, Beginning PythonFrom Novice to Professional, APress, 2005
6. Mark Lutz, Programming Python, O'Reilly Media, 2011
7. https://www.python.org/

8.2 Activităţi aplicative[13]	Număr de ore	Metode de predare
1. Norme de tehnica securităţii muncii. Reprezentarea de algoritmi prin intermediul schemelor logice care utilizează structura secvenţială, structura alternativă și structura repetitivă	4	Se va utiliza exercițiul la tablă.
2. Reprezentarea de algoritmi prin intermediul limbajului pseudocod utilizând cele trei structuri de control	4	
3. Fixarea cunoștințelor. Reprezentarea algoritmilor folosind schemelor logice și limbajul pseudocod	4	
4. Rezolvarea problemelor utilizând subalgoritmi implementați iterativ și/sau recursiv	4	
5. Implementarea în Python a programelor corespunzătoare algoritmilor care utilizează instrucțiunile specifice celor trei structuri de control	8	Se va utiliza exercițiul la tablă și implementa-rea programului utilizând computerul
6. Utilizarea funcțiilor la implementarea programelor. Recursivitate	4	Se va utiliza exercițiul la tablă și implementa-rea programului utilizând computerul
7. Implementarea în Python a programelor corespunzătoare algoritmilor care utilizează tablouri unidimensionale și bidimensionale	8	Se va utiliza exercițiul la tablă și implementa-rea programului utilizând computerul

8. Utilizarea tipurilor de date compuse la implementarea programelor	4	Se va utiliza exercițiul la tablă și implementarea programului utilizând computerul
9. Implementarea în Python a programelor ce utilizează fișiere text și fișiere binare	2	Se va utiliza exercițiul la tablă și implementarea programului utilizând computerul

Bibliografie[14]

1. Ghiormez L., Laborator în format electronic – Programarea calculatoarelor și limbaje de programare 1, 2026, Campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=3546 sau intranet facultate \\fs\cursuri\PCLP 1 2026
2. Anca-Elena Iordan, Daniela Cristea, Adela Berdie, Programarea calculatoarelor, Editura Mirton, Timişoara, 2004
3. Robert Sedgewick, Kevin Wayne, Algorithms, Pearson Education, 2011
4. Magnus Lie Hetland, Python Algorithms Mastering Basic Algorithms in the Python Language, APress, 2010
5. David Beazley, Python Essential Reference, Sams Publishing, 2006
6. https://pythonprogramming.net/python-fundamental-tutorials/
7. https://www.python.org/

9. Evaluare

Tip activitate	**9.1** Criterii de evaluare[15]	**9.2** Metode de evaluare	**9.3** Pondere din nota finală
9.4 Curs	Cunoștințe teoretice	Scris - subiecte teoretice și aplicații. Studenții au acces la computere, dar rezolvările se notează pe foaie. Examenul se desfășoară prin intermediul Campusului Virtual UPT sau se primesc subiecte tipărite. În caz de scenariu online se utilizează și aplicația de videoconferință (Zoom), iar rezolvările se încarcă pe campusul virtual UPT	0.66
9.5 Activităţi aplicative	**S:**		
	L: Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator. Teme de casă și/sau în timpul desfășurării activității practice	Evaluări pe parcursul semestrului: - Scris – aplicații folosind scheme logice/ limbaj pseudocod și limbaj Python. - Oral – aplicații utilizând computerul. In caz de scenariu online testele de control se desfășoară prin intermediul campusului virtual UPT	0.34
	P[16]**:**		
	Pr:		

9.6 Standard minim de performanţă (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor[17])

- **Laborator***:* Studentul trebuie să dețină capacitatea de a extrage datele problemei, de a utiliza documentația furnizată și de a rezolva corect o problemă concretă folosind structura alternativă imbricată reprezentată prin intermediul schemelor logice sau a limbajului pseudocod și a limbajului de programare Python.

- **Curs**:
  - Studentul trebuie să identifice și să utilizeze elementele grafice de bază ale unei scheme logice și structura corespunzătoare în pseudocod, respectiv în limbajul Python pentru un algoritm simplu bazat pe structura alternativă imbricată.
  - Studentul trebuie să identifice corect datele de intrare, datele intermediare și datele de ieșire, să definească o condiție simplă pentru a rezolva o problemă concretă de complexitate redusă și să fie capabil să scrie, să ruleze și să depaneze în mediul de dezvoltare (IDE) un script simplu (fără utilizarea colecțiilor de date) Python funcțional.
  - Studentul trebuie să fie capabil să rezolve o problemă concretă pe baza documentației tehnice de bază (ex. python.org) sau a exemplelor prezentate la curs sau la laborator, evitând plagiatul de cod.

- *Pentru a promova disciplina trebuie ca ambele activități (curs și laborator) să fie promovate cu minim 5.*

Data completării	**Titular de curs (semnătura)**	**Titular activităţi aplicative (semnătura)**
10.09.2026		

Director de departament (semnătura)	Data avizării în Consiliul Facultăţii[18]	Decan (semnătura)
	23.09.2026	