

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Facultatea de Științe (Centrul Universitar Nord din Baia Mare)
1.3 Departamentul	Departamentul de Matematică și Informatică
1.4 Domeniul de studii	Matematică
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Matematică informatică (Baia Mare, limba română) / Licențiat în matematică
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Algoritmi fundamentali și structuri de date I	Codul disciplinei	4.00
2.2 Titularul de curs	Conf. univ. dr. Cosmin Nicolae SABO – cosmin.sabo@mi.utcluj.ro		
2.3 Titularul activităților de seminar / laborator / proiect / practică	Asistent universitar, post vacant		
2.4 Anul de studiu	I	2.5 Semestrul	1
		2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categoriza formativă		DF
	Opționalitate		DOB

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	3.2 Curs	2	3.3 Seminar	-	3.3 Laborator	2	3.3 Proiect	-	3.3 Practică	-
3.4 Număr de ore pe semestru	56	din care:	3.5 Curs	28	3.6 Seminar	-	3.6 Laborator	28	3.6 Proiect	-	3.3 Practică	-
3.7 Distribuția fondului de timp (ore pe semestru) pentru studiu individual și evaluare:												
(a) Evaluare												6
(b) Studiul după manual, suport de curs, bibliografie și notițe												20
(c) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren												8
(d) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri												30
(e) Tutoriat												3
(f) Alte activități												2
3.8 Total ore studiu individual și evaluare (suma (3.7(a))...3.7(f))								69				
3.9 Total ore pe semestru (3.4+3.8)								125				
3.10 Numărul de credite								5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Nu este cazul (disciplina se predă în semestrul 1 al anului I).
4.2 de competențe	Matematică de nivel liceal (operații cu numere, procente, logică elementară) și competențe digitale de bază (utilizarea unui sistem de operare, a unui browser și a unui editor de text). Nu sunt necesare cunoștințe de programare; disciplina pornește de la zero și include un chestionar de diagnostic în prima săptămână.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs cu videoproiector și acces la internet; cursul îmbină aspectele teoretice cu exemple și demonstrații, pe care studenții le pot urmări și rula pe laptopul personal. Suportul de curs, exemplele rezolvate, testele de autoevaluare și anunțurile se publică pe platforma Campus Virtual; studenții au cont instituțional activ din prima săptămână. Cursul este comun programelor Matematică informatică, Informatică și Informatică economică.
5.2. de desfășurare a seminarului / laboratorului / proiectului	Laborator cu calculatoare pe care sunt instalate interpretorul limbajului de programare utilizat (Python, versiune curentă) și un editor de cod; se recomandă lucrul pe laptopul personal, cu același mediu instalat în săptămâna 1. Activitatea este individuală: fiecare student continuă la laborator problema săptămânii începută la curs, rezolvă sarcinile de laborator și realizează proiectul final; livrabilele se predau prin Campus Virtual până la termenele anunțate și sunt verificate automat, cu feedback la laborator. Asistenții bazați pe inteligență artificială sunt permisi la laborator și la proiect, cu declararea lor în fiecare livrabil și cu obligația de a explica propriul cod; sunt interziși la teste și la examen. Prezența la laborator este obligatorie, conform regulamentului UTCN privind activitatea profesională a studenților.

6. Competențele specifice acumulate

Competențe profesionale	C5. Gândește în mod abstract: modelează o problemă concretă prin date și pași de prelucrare (algoritm), independent de limbaj. C10. Găsește soluții pentru probleme: analizează enunțul, construiește exemple, proiectează și implementează o soluție corectă și o testează. C11. Dezvoltă strategii de soluționare a problemelor: alege între algoritmi alternativi de căutare și sortare pe criteriul complexității și al dimensiunii datelor. C15. Utilizează tehnici de prelucrare a datelor: prelucrează date din fișiere text și tabelare și produce rapoarte simple.
Competențe transverale	CT1. Gândește critic: verifică soluțiile pe cazuri de test, identifică și corectează erori de raționament și de implementare. CT3. Respectă angajamente: predă lucrările de laborator și proiectul la termen, în forma cerută. CT5. Gestionează evoluția personală: se documentează autonom din surse recomandate și folosește responsabil instrumentele de asistență.

7. Rezultatele așteptate ale învățării

Cunoștințe	RI1. Descrie noțiunile de algoritm și program, etapele rezolvării unei probleme (înțelegere, exemple, proiectare, implementare, testare) și elementele de bază ale limbajului: tipuri de date, expresii, structuri de control, funcții, colecții (șiruri de caractere, liste, tupluri, dicționare, mulțimi, matrice). RI2. Explică algoritmi fundamentali de căutare (secvențială, binară) și de sortare (prin interschimbare, selecție, inserție), recursivitatea și tipurile abstracte de date stivă și coadă, cu proprietățile și aplicațiile lor. RI3. Explică noțiunea de complexitate a unui algoritm, notația O și clasele uzuale de complexitate și compară doi algoritmi pentru aceeași problemă după numărul de operații.
------------	---

Abilitați	RI4. Scribe, testează și depanează programe corecte și lizibile care rezolvă probleme date, folosind variabile, structuri de control, funcții și colecții, și organizează codul în funcții cu nume și responsabilități clare. RI5. Implementează algoritmi fundamentali de căutare și sortare, iterativ și recursiv, și alege între ei în funcție de dimensiunea datelor și de complexitate. RI6. Prelucreează date din fișiere text și tabelare (CSV), cu tratarea erorilor, și produce rapoarte simple. RI7. Trasează manual execuția unui program pe un exemplu, estimează complexitatea unui algoritm simplu și măsoară experimental timpul de execuție.
Responsabilitate și autonomie	RI8. Rezolvă autonom probleme de dificultate crescătoare, își verifică soluțiile pe cazuri de test proprii și își planifică lucrul astfel încât să respecte termenele de predare. RI9. Utilizează responsabil sursele de documentare și instrumentele de asistență, inclusiv cele bazate pe inteligență artificială, le declară și este capabil să explice fiecare linie din propriul cod.

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

8.1 Obiectivul general al disciplinei	Formarea gândirii algoritmice și a deprinderii de a transforma o problemă concretă într-un program corect, lizibil și testat, prin exemple rezolvate și exersare săptămânală, într-un mod accesibil studenților fără experiență anterioară de programare.
8.2 Obiectivele specifice	O1. Înțelegerea noțiunilor de algoritm, program și date și a etapelor rezolvării unei probleme. O2. Stăpânirea elementelor de bază ale unui limbaj de programare de nivel înalt: tipuri de date, expresii, structuri de control, funcții, șiruri de caractere, liste, tupluri, dicționare, mulțimi și matrice. O3. Implementarea și compararea algoritmilor fundamentali de căutare și sortare, iterativ și recursiv. O4. Înțelegerea intuitivă și experimentală a complexității algoritmilor. O5. Utilizarea tipurilor abstracte de date stivă și coadă și prelucrarea datelor din fișiere. O6. Formarea deprinderilor de testare, depanare, stil de cod și de utilizare responsabilă a instrumentelor de asistență.

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
1. Introducere: algoritm, program, gândire algoritmică; instrumentele de lucru; primul program.	2	Prelegere interactivă cu aspectele teoretice și exemple; întrebări și sondaje; suport de curs, exemple și teste de autoevaluare pe Campus Virtual.	Tema cursului continuă la laborator. Teste scurte în săpt. 6 și 11. Se cere nucleul din suport.
2. Variabile, tipuri de date de bază, expresii; citirea și afișarea datelor; conversii.	2		
3. Structuri de control: decizii și repetiții; trasarea execuției.	2		
4. Șiruri de caractere: indexare, feliere, metode, formatare; stil de cod.	2		
5. Funcții: parametri, rezultat, domeniu de vizibilitate; descompunerea problemelor; teste simple.	2		
6. Liste și tupluri: operații, parcurgeri, construcții compacte; algoritmi elementari pe secvențe.	2		
7. Căutarea secvențială și binară, iterativ și recursiv; introducere în recursivitate.	2		
8. Sortarea elementară: prin interschimbare, selecție, inserție; stabilitate; sortarea încorporată.	2		
9. Complexitatea algoritmilor: numărarea operațiilor, notația O, clase uzuale; măsurare experimentală.	2		
10. Dicționare și mulțimi: operații, frecvențe, grupări; module standard.	2		
11. Matrice ca liste de liste: parcurgeri, transpunere, operații pe submatrice.	2		
12. Fișiere text și tabelare (CSV); tratarea erorilor.	2		
13. Tipuri abstracte de date: stivă și coadă; implementare cu liste; aplicații.	2		
14. Sinteză: de la problemă la aplicație (studiu de caz integrator); pregătirea examenului.	2		
Bibliografie			
1. Suportul de curs al disciplinei, publicat pe Campus Virtual.			
2. Downey, A. B., Think Python: How to Think Like a Computer Scientist, ed. a 3-a, O'Reilly, 2024; ediție online gratuită, https://alldowney.github.io/ThinkPython/ .			
3. Python Software Foundation, The Python Tutorial (există și traducere în limba română), https://docs.python.org/3/tutorial/ .			
4. Swaroop, C. H., A Byte of Python (disponibilă și în traducere românească), https://python.swaroopch.com .			
5. Miller, B. N., Ranum, D. L., Problem Solving with Algorithms and Data Structures using Python, ed. a 3-a, Runestone Academy; ediție interactivă gratuită, https://runestone.academy/ns/books/published/pythonds3/ .			
6. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., Introduction to Algorithms, ed. a 4-a, MIT Press, 2022 (cap. 1-3, referință).			
7. Giumale, C., Introducere în analiza algoritmilor: teorie și aplicație, Polirom, 2004.			
8. Sedgewick, R., Wayne, K., Dondero, R., Introduction to Programming in Python: An Interdisciplinary Approach, Addison-Wesley, 2015; materiale gratuite, https://introcs.cs.princeton.edu/python .			
9. Culegeri de probleme cu evaluare automată în limba română: https://www.pbinfo.ro , https://infoarena.ro .			

9.2 Seminar / laborator / proiect / practică	Nr. ore	Metode de predare	Observații
1. Instalarea mediului de lucru; primul script; lucrul în consolă; chestionar de diagnostic.	2	Lucru individual pe calculator, pornind de la problema săptămânii începută la curs; sarcini pe două niveluri (de bază și extindere), verificate automat, cu feedback la laborator; fișe de laborator și soluții de referință pentru titularul de laborator; asistenți AI permiși cu declarare; predare prin Campus Virtual.	Pe subgrupe.
2. Probleme aritmetice cu date citite de la tastatură și afișare formatată.	2		Lucrări notate L1-L6 (săpt.
3. Tabele de trasare; probleme cu decizii și bucle (L1).	2		3, 5, 7, 9,
4. Validarea datelor introduse: calculatorul text.	2		11, 13).
5. Jocul „Ghicește numărul”, organizat pe funcții (L2).	2		Proiect final în
6. Gestiunea notelor studenților cu liste; test scurt T1 (săpt. 1-5).	2		săpt. 12-14.
7. Agendă telefonică cu căutare secvențială și binară (L3).	2		Nivelare în săpt. 1-4 pentru
8. Vizualizator în mod text al algoritmilor de sortare; clasament după mai multe criterii.	2		începători .
9. Măsurarea timpului de execuție și compararea algoritmilor; grafic în mod text (L4).	2		
10. Dicționar interactiv cu operații de adăugare, căutare, modificare și ștergere.	2		
11. Jocul X și 0 pe matrice (L5); test scurt T2 (săpt. 6-10).	2		
12. Analizor de date dintr-un fișier CSV, cu raport; anunțarea temelor proiectului final.	2		
13. Evaluator de expresii cu stivă; simularea unei cozi (L6).	2		
14. Prezentarea și evaluarea proiectului final.	2		
Bibliografie			
1. Îndrumarul de laborator al disciplinei (fișe de laborator, șabloane de cod, teste automate, temele proiectului), publicat pe Campus Virtual.			
2. Python Software Foundation, The Python Standard Library, https://docs.python.org/3/library/ .			
3. van Rossum, G., Warsaw, B., Coghlan, A., PEP 8: Style Guide for Python Code, https://peps.python.org/pep-0008/ .			
4. Miller, B. N., Ranum, D. L., Problem Solving with Algorithms and Data Structures using Python, ediția interactivă, https://runestone.academy/ns/books/published/python3/ .			
5. Culegeri de probleme cu evaluare automată: https://www.pbinfo.ro , https://infoarena.ro .			

10. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

Algoritmii fundamentali și structurile de date sunt nucleul comun al tuturor programelor de informatică și matematică informatică din țară și din străinătate și al recomandărilor curriculare internaționale (ACM/IEEE-CS Computer Science Curricula 2023, ariile „Software Development Fundamentals” și „Algorithmic Foundations”). Competențele formate (rezolvarea de probleme prin programare, alegerea algoritmului potrivit, testarea și depanarea, prelucrarea datelor din fișiere) sunt cele verificate de angajatorii din industria software la angajarea absolvenților și a studenților în stagii de practică, inclusiv de firmele din regiune cu care departamentul colaborează prin programele de practică și prin întâlnirile periodice cu mediul economic. Ele corespund ocupațiilor din ESCO/COR vizate de programe: programator de sistem informatic (251204) pentru Informatică, matematician și consilier statistician (212009, 212011) pentru Matematică informatică, precum și ocupațiilor din informatica economică.

Disciplina este baza pentru Algoritmi fundamentali și structuri de date II, Dezvoltare colaborativă și tehnologii cloud (același semestru, fluxul de lucru cu cod versionat), Inteligență Artificială Generativă

Aplicată, Programare orientată pe obiecte, Baze de date și Tehnici avansate de programare, iar abordarea prin probleme rezolvate și evaluare continuă urmează practica cursurilor introductive de referință din comunitatea academică.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Cunoașterea și înțelegerea conceptelor (RI1-RI3): elementele limbajului, algoritmi fundamentali de căutare și sortare, recursivitatea, stiva și coada, complexitatea. Capacitatea de a rezolva individual, pe calculator, probleme noi de dificultate graduală, cu cod corect, lizibil și testat (RI4, RI5, RI7).	Evaluare continuă: două teste scurte pe Campus Virtual (T1 în săpt. 6, materia săpt. 1-5; T2 în săpt. 11, materia săpt. 6-10), corectate automat (10%). Evaluare sumativă, examen în sesiune: probă practică individuală pe calculator, 2 ore, trei probleme de dificultate graduală, fără instrumente de asistență, verificată automat și citită de titular (30%); chestionar conceptual pe Campus Virtual, 30 de minute (10%).	50% (10% teste scurte, 30% probă practică, 10% chestionar conceptual)
11.5 Seminar/Laborator /Proiect / practică	Portofoliul de laborator (RI4-RI7, RI9): sarcinile săptămânale cu nivelul de bază trecut de verificarea automată și lucrările L1-L6, cu cod corect, lizibil și testat; respectarea termenelor; declararea instrumentelor de asistență. Proiectul final (RI4, RI6, RI8, RI9): aplicație individuală mică, funcțională în cerințele de bază, cu date în fișiere, organizată pe funcții, prezentată și explicată la laborator.	Evaluare continuă: portofoliul de laborator, predat prin Campus Virtual, verificat automat și notat pe rubrică de titularul de laborator (30%). Evaluare continuă: proiectul final, pe una dintre temele date, predat în săpt. 14 și prezentat la laborator, notat pe rubrică; studentul explică și modifică la cerere propriul cod (20%).	50% (30% portofoliul de laborator, 20% proiectul final)

11.6 Standard minim de performanță

Nota finală $N = 0,10 \times TS + 0,30 \times PL + 0,20 \times PF + 0,30 \times EP + 0,10 \times EC$, unde TS = media testelor scurte, PL = portofoliul de laborator, PF = proiectul final, EP = proba practică, EC = chestionarul conceptual, toate notate de la 1 la 10. Evaluarea continuă (TS, PL, PF) reprezintă 60% din nota finală, evaluarea sumativă (EP, EC) 40%.

Condiții minime pentru nota 5: toate cele șase lucrări L1-L6 predate și $PL \geq 5$; proiectul final predat și funcțional în cerințele de bază, explicat de student; ambele teste scurte susținute; la proba practică, studentul rezolvă singur, complet și corect, cel puțin o problemă și obține $EP \geq 5$; nota finală $N \geq 5$.

Livrabilele întârziate pierd un punct pe săptămână din nota livrabilului. În sesiunea de restanțe se reface numai examenul (proba practică și chestionarul conceptual); testele, portofoliul și proiectul rămân valabile în anul universitar curent. Instrumentele de asistență se declară în fiecare livrabil și sunt interzise la teste și la examen; nedeclararea sau incapacitatea de a explica propriul cod se tratează ca abatere de la integritatea academică, conform regulamentelor UTCN.

Data completării:	Titulari	grad didactic, titlu Prenume NUME	Semnătura
10.09.2026	Curs	Conf. univ. dr. Cosmin Nicolae SABO	
	Aplicații	Asistent universitar, post vacant	
		-	

Avizul Directorului departamentului care coordonează disciplina, doar dacă acesta diferă de departamentul organizator al programului de studii

Nu este cazul (disciplina este coordonată de departamentul organizator al programului de studii).

Data avizării în Consiliul Departamentului: 14.09.2026

Director Departament DMI
Conf. univ. dr. Cosmin Nicolae SABO

Data aprobării în Consiliul Facultății: 17.09.2026

Decan,
Conf. univ. dr. Monica Liliana MARIAN