

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Transilvania din Brașov
1.2. Facultatea	Matematică și Informatică
1.3. Departamentul	Matematică și Informatică
1.4. Domeniul de studii de licență	Matematică
1.5. Ciclul de studii ²⁾	Licență
1.6. Programul de studii / Calificarea	Matematică informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	ALGORITMI ȘI PROGRAMARE							
2.2 Titularul activităților de curs	Lect. Dr. Luciana Majercsik							
2.3 Titularul activităților de seminar/ laborator/ proiect	Drd. Dumbravă Claudiu							
2.4 Anul de studiu	I	2.5 Semestrul	I	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Conținut ²⁾	DF
							Obligativitate ³⁾	DI

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	5	din care: 3.2 curs		2	3.3 seminar/ laborator/ proiect	1/2/0
3.4 Total ore din planul de învățământ	70	din care: 3.5 curs		28	3.6 seminar/ laborator/ proiect	14/28/0
Distribuția fondului de timp						ore
Studiul după manual, suport de curs, bibliografie și notițe						30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren						12
Pregătire seminarii/ laboratoare/ proiecte, teme, referate, portofolii și eseuri						30
Tutoriat						6
Examinări						2
Alte activități.....						
3.7 Total ore studiu individual	80					
3.8 Total ore pe semestru	150					
3.9 Numărul de credite	5					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Noțiuni elementare din liceu: variabile, constante, instrucțiuni, subprograme. • Algoritmi elementari de lucru cu vectori și matrice
4.2 de competențe	• Competențe informatice/digitale la nivel preuniversitar

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	Sală de minim 50 de locuri, videoproiector
-------------------------------	--

5.2 de desfășurare a seminarului/ laboratorului/ proiectului	• Instalat Visual Studio IDE • Instalat NotePad++ • Internet
--	--

6. Competențe specifice acumulate

Competențe profesionale	Cp3. Elaborarea și analiza unor algoritmi pentru rezolvarea problemelor, precum și identificarea de modele statistice 3.1. Cunoștințe R.Î.3.1.1. Absolventul definește conceptele din disciplinele de bază de informatică și/sau matematici aplicate. R.Î.3.1.2. Absolventul compară și distinge noțiunile înrudite și proprietățile acestora din disciplinele de bază de informatică și/sau matematici aplicate. 3.2. Aptitudini R.Î.3.2.1. Absolventul identifică și aplică tehnicile adecvate pentru rezolvarea exercițiilor și problemelor din disciplinele majore ale matematicii. R.Î.3.2.2. Absolventul identifică și aplică tehnicile adecvate pentru rezolvarea problemelor din disciplinele avansate de matematică. R.Î. 3.1. Absolventul interpretează rezultatele prelucrării datelor folosind tehnologii specifice. R.Î. 3.2. Absolventul identifică noțiunile legate de algoritmi folosind tehnologii specifice. R.Î. 3.3. Absolventul elaborează coduri sursă într-un limbaj de programare folosind tehnologii specifice. 3.3. Responsabilitate și autonomie R.Î.3.3.1. Absolventul identifică și corelează legături între concepte aparent fără legătură din disciplinele majore ale matematicii. R.Î.3.3.2. Absolventul rezumă, clasifică și prezintă concluziile unor probleme date folosind diverse tipuri de reprezentări și comunică clar și eficient concepte și raționamente matematice la specialiști și nespecialiști prin rapoarte scrise și prezentări orale. R.Î. 3.4. Absolventul analizează datele statistice pentru a găsi modele și tendințe în date sau între variabile.
	Cp5. Utilizarea software-ului pentru design specializat, dezvoltarea de software cu sursă deschisă 5.1. Cunoștințe R.Î. 5.1. Absolventul dezvoltă noi modele prin utilizarea abilității a unui software specializat. 5.2. Aptitudini R.Î. 5.2. Absolventul exploatează și creează software cu sursă deschisă. Este familiarizat(ă) cu principalele modele de software cu sursă deschisă, cu sistemele de acordare a licențelor și cu practicile de codificare adoptate în mod obișnuit în producția de software cu sursă deschisă. 5.3. Responsabilitate și autonomie R.Î. 5.3. Absolventul poate să descrie elemente structurale ale limbajelor de programare folosind limbajul specific. R.Î. 5.4. Absolventul poate să explice funcționarea mediilor de programare folosind limbajul specific. R.Î. 5.5. Absolventul poate să dezvolte programe pe baza unor specificații date folosind tehnologii specifice.

Competențe transversale	CT2. Competențe digitale: comunicare, colaborare și utilizarea software-ului de bază Rezultate ale învățării RÎ 2.1: Absolventul utilizează instrumente și tehnologii digitale simple pentru a comunica, a interacționa și a colabora cu ceilalți. RÎ 2.2: Absolventul realizează și gestionează una sau mai multe identități digitale, își protejează propria reputație și gestionează datele personale create în medii și servicii digitale diverse. RÎ 2.3: Absolventul scrie și aplică instrucțiuni simple pentru un sistem informatic, în vederea rezolvării unor probleme sau a îndeplinirii unor sarcini de bază, cu sprijin orientativ, dacă este necesar.
-------------------------	--

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

7.1 Obiectivul general al disciplinei	Însușirea unei gândiri algoritmice și dezvoltarea abilităților de proiectare de soluții algoritmice, precum și însușirea tehnicilor de utilizare a principalelor structuri de date.
7.2 Obiectivele specifice	• inițiere în utilizarea unui limbaj de programare • implementarea principalelor structuri de date • însușirea tehnicilor de bază în proiectarea programelor • consolidarea deprinderilor de a justifica și evalua complexitatea algoritmilor • evaluarea timpului de execuție a algoritmilor în cazurile favorabil/mediu/defavorabil • dezvoltarea abilităților de proiectare de soluții algoritmice

8. Conținuturi

8.1 Curs	Metode de predare	Observații
Noțiuni introductive. Componentele unui algoritm. Pseudocod și convenții. Variabile.	Prelegeri	2 ore
Analiza eficienței algoritmilor iterativi. Calcul de complexități pe diverse probleme.		6 ore
Algoritmi de căutare. Discuții legate de complexități		2 ore
Algoritmi de sortare. Discuții legate de complexități.	Expuneri	2 ore
Algoritmi Divide et impera	Dialog	2 ore
Algoritmi de tip Greedy	Problematizare	4 ore

Algoritmi de programare dinamică	Studiu de caz	4 ore
Algoritmi pe grafuri		2 ore
Algoritmi numerici		4 ore
	Exemple	
		Total 28 de ore
Bibliografie: - <i>Algoritmi fundamentali. O perspectivă C++</i> - Andonie R., Gârbacea I., Ed. Libris, Cluj-Napoca, 1995 - <i>Introduction to Algorithms</i> - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press, 2001 - <i>Programming Fundamentals</i> - Frențiu M., Pop H.F., Șerban G., Ed. Presa Universitară Clujeană Cluj-Napoca, 2006 - <i>Algorithms Unlocked</i> – Thomas H. Cormen, MIT Press, 2013 - <i>The Algorithm Design Manual</i> – Steven S. Skiena, Springer, 2020 (3rd Edition) - <i>Computational Complexity: A Modern Approach</i> – Sanjeev Arora, Boaz Barak, Cambridge University Press, 2009 - <i>Algorithm Design: Foundations, Analysis, and Internet Examples</i> – Michael T. Goodrich, Roberto Tamassia, Wiley, 2002		
8.2 Seminar/ laborator/ proiect	Metode de predare-învățare	Observații
Date cu care lucrează algoritmi (constante, variabile, operanzi/operatori, identificatori) Algoritmi elementari de procesare de numere. Reprezentarea algoritmilor în pseudocod și limbaj de programare (C/C++) Tablouri unidimensionale și bidimensionale	Exerciții	8 ore
Algoritmi elementari transpuși utilizând tablouri unidimensionale, bidimensionale și structuri (struct/class)		2 ore
Analiza comparativă a implementărilor și a metodelor de programare		4 ore
Algoritmi de căutare și sortare	Dialog	4 ore
Algoritmi Divide et impera		6 ore
Algoritmi de tip Greedy	Problematizare	6 ore
Algoritmi de programare dinamică		6 ore
Algoritmi pe grafuri		2 ore
Algoritmi numerici	Studiu individual	2 ore

Discuții și implementări recapitulative, pregătitoare pentru examen		2 ore
		Total 42 de ore
Bibliografie: - <i>Algoritmi fundamentali. O perspectivă C++</i> - Andonie R., Gârbacea I., Ed. Libris, Cluj-Napoca, 1995 - <i>Introduction to Algorithms</i> - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press, 2001 - <i>Programming Fundamentals</i> - Frențiu M., Pop H.F., Șerban G., Ed. Presa Universitară Clujeană Cluj-Napoca, 2006 - <i>Algorithms Unlocked</i> – Thomas H. Cormen, MIT Press, 2013 - <i>The Algorithm Design Manual</i> – Steven S. Skiena, Springer, 2020 (3rd Edition) - <i>Computational Complexity: A Modern Approach</i> – Sanjeev Arora, Boaz Barak, Cambridge University Press, 2009 - <i>Algorithm Design: Foundations, Analysis, and Internet Examples</i> – Michael T. Goodrich, Roberto Tamassia, Wiley, 2002		

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

Fișa disciplinei a fost elaborată după consultări cu partenerii din mediul academic și economic ai Facultății.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
10.4 Curs	Activitate continuă și participare la curs	Prezență activă și participare la discuții. Integrarea conceptelor teoretice (algoritmi, metode de programare, complexitate de timp și spațiu etc) în exemple și dezbateri. Demonstrarea unei gândiri critice și reflexive asupra strategiilor algoritmice prezentate. Evaluare continuă.	10%
10.5 Seminar/ laborator/ proiect	Activitate continuă și participare la laborator	Evaluare continuă cu notă a temelor de laborator + activității în timpul orelor de laborator, cu pondere de 20% din nota de laborator Evaluare cu notă în cadrul unei testări de laborator la sfârșitul semestrului, cu pondere de 80 % din nota de laborator.	40%
Examen final		Proba scrisă care constă în grile și probleme bazate pe teorie.	50%

		Promovarea examenului presupune obținerea de minim nota 5 la această probă.	
10.6 Standard minim de performanță			
• Elaborare algoritmi elementari (procesare de numere, tablouri unidimensionale și bidimensionale) • Calcul de complexități pentru algoritmi elementari • Algoritmi de căutare și sortare • Algoritmi elementari de divide et impera (principiile de bază și elaborare de algoritmi) • Algoritmi elementari de tip Greedy (principiile de bază și elaborare de algoritmi) Studentul trebuie să demonstreze, atât în sarcinile practice de laborator, cât și în răspunsurile teoretice la curs, că înțelege și poate aplica coerent aceste competențe. Nedemonstrarea capacității de a răspunde logic și coerent la întrebările legate de implementare atrage evaluarea la nivel insuficient.			

Prezenta Fișă de disciplină a fost avizată în ședința de Consiliu de departament din data de 15/09/2025 și aprobată în ședința de Consiliu al facultății din data de 15/09/2025.

Decan Conf. Dr. Ion Gabriel Stan	Director de departament Conf. Dr. Nicușor Minculete
Titular de curs Lect. Dr. Luciana Majercsik	Titular de laborator Drd. Dumbravă Claudiu

Notă:

- ¹⁾ Domeniul de studii - se alege una din variantele: Licență/ Masterat/ Doctorat (se completează conform cu Nomenclatorul domeniilor și al specializărilor/ programelor de studii universitare în vigoare);
- ²⁾ Ciclul de studii - se alege una din variantele: Licență/ Masterat/ Doctorat;
- ³⁾ Regimul disciplinei (conținut) - se alege una din variantele: **DF** (disciplină fundamentală)/ **DD** (disciplină din domeniu)/ **DS** (disciplină de specialitate)/ **DC** (disciplină complementară) - pentru nivelul de licență; **DAP** (disciplină de aprofundare)/ **DSI** (disciplină de sinteză)/ **DCA** (disciplină de cunoaștere avansată) - pentru nivelul de masterat;
- ⁴⁾ Regimul disciplinei (obligativitate) - se alege una din variantele: **DI** (disciplină obligatorie)/ **DO** (disciplină opțională)/ **DFac** (disciplină facultativă);
- ⁵⁾ Un credit este echivalent cu 30 de ore de studiu (activități didactice și studiu individual).