

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. Ciclu de studii ²⁾	Licență
1.6. Programul de studii / Calificarea	Matematică informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Structuri de date							
2.2 Titularul activităților de curs	Lect. dr. Plajer Ioana Cristina							
2.3 Titularul activităților de laborator	Asist. drd. Popa Iulian							
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Conținut ³⁾	DS
							Obligativitate ³⁾	DI

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

3.1 Număr de ore pe săptămână	3	din care: 3.2 curs	2	3.3 seminar/ laborator/ proiect	0/1/0
3.4 Total ore din planul de învățământ	42	din care: 3.5 curs	28	3.6 seminar/ laborator/ proiect	0/14/0
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					10
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					30
Tutorat					4
Examinări					4
Alte activități.....					0
3.7 Total ore studiu individual	78				
3.8 Total ore pe semestru	120				
3.9 Numărul de credite ⁵⁾	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Preferabil limbaj programare C, cunoștințe de matematică de nivel de liceu - profil real, cunoștințe minime de engleză
4.2 de competențe	Preferabil elaborarea unor programe în C, capacitatea de a implementa algoritmi din pseudo-cod

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	Tablă, proiector, laptop
5.2 de desfășurare a seminarului/ laboratorului/ proiectului	Calculatoare, medii de programare C/C++ - Visual Studio / Code Blocks, tablă

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 2.3: 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 competențele specifice acumulate)

7.1 Obiectivul general al disciplinei	Cunoașterea și înțelegerea unor structurilor de date de bază necesare pentru în dezvoltarea aplicațiilor software.
7.2 Obiectivele specifice	- Dezvoltarea capacității studenților de implementa structuri de date de bază, precum și de a le aplica în contextul rezolvării de probleme. - Dezvoltarea abilității studenților de a interpreta și evalua rezultatele produse de propriile programe, de a estima eficiența lor și de a alege structuri potrivite în contexte potrivite.

8. Conținuturi

8.1 Curs	Metode de predare	Număr de ore	Observații
1. Introducere, limbajul pseudo-cod.		2	

2. Tablouri unidimensionale. Tablouri bidimensionale. Algoritmi. Noțiuni de complexitate. Implementare C++.	Expunere	6	
3. Funcții. Structuri folosind struct.	Dialog		
4. Structuri elementare: 1.1. Liste înlanțuite 1.2. Stiva 1.3 Coda	Explicație	2	
5. Biblioteca STL. Structuri de bază (pair, tuple, vector, stack, queue, list)	Dezbateri	4	
6. Tabele de dispersie	Informare	2	
7. Noțiuni de grafuri, arbori rădăcină: 7.1 Grafuri: noțiuni elementare, reprezentare 7.2 Arbori oarecare: definiții, reprezentare, parcurgere		2	
8. Heap. Coadă de priorități		2	
9. Arbori binari de căutare. Arbori Roșu-Negru		4	
10. Sinteză și recapitulare pentru examen	Dialog.	2	

Bibliografie

1. Cormen T.H, Leiserson C.E, Rivest R., Clifford S. „Intorduction to Algorithms” fourth Edition, MIT Press 2022
2. NICULESCU V, CZIBULA, G.:”Structuri fundamentale de date. O perspectivă orientată obiect.” Editura Casa Cartii de Stiinta, Cluj - Napoca, 2011
3. Plajer I. C., "Noțițe de curs pentru structuri de date". 2024
4. Cherchez E., Marinel S., "Programarea în limbajul C/C++ pentru liceu", volumul 3, editura Polirom 2006
5. Weiss, M.A., "Data Structures and Algorithm Analysis in C++ ", fourth edition, Pearson 2014

8.2 Laborator	Metode de predare- învățare	Număr de ore	Observații
1. Probleme elementare în C++.	Lucrări	2	
2. Vectori. Matrice.	practice/Aplicații C++.	3	
3. Stive. Cozi. Liste înlanțuite. Grafuri	Exerciții.	3	
4. Heap-uri. Arbori binari de căutare.	Rezolvare exerciții.	4	
	Algoritmizare.		
5. Evaluare finală	. Testare finală	2	

Bibliografie

1. Cormen T.H, Leiserson C.E, Rivest R., Clifford S. "Introducere în algoritmi", ediția a II-a, editura Byblos 2004
2. Cormen T.H, Leiserson C.E, Rivest R., Clifford S. „Intorduction to Algorithms” fourth Edition, MIT Press 2022
3. Cherchez E., Marinel S., "Programarea în limbajul C/C++ pentru liceu", volumul 3, editura Polirom 2006
4. Plajer I. C., "Noțițe de curs pentru structuri de date". 2024
5. A. Allain "Jumping into C++", 2013

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	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Activitate continuă și prezență la curs.	Prezență activă și participare la discuții. Însușirea structurilor de bază și înțelegerea contextelor de aplicabilitate și a complexității operațiilor. Evaluare continuă prin testări sub formă de quiz-uri.	10 %
	Probă scrisă: Cunoașterea noțiunilor; Înțelegerea structurilor de date și a proprietăților lor; Capacitatea de aplicare a cunoștințelor în exerciții concrete.	Probă scrisă (grile + probleme bazate pe teorie) în sesiunea de examene. Baremul de notare este transmis studenților odată cu subiectele. Promovarea examenului presupune obținerea de minim nota 5 la această probă.	65%
10.5 Laborator	Activitate continuă și prezență la laborator Capacitatea de implementare a structurilor de date învățate precum și de utilizare adecvată a acestora în contextul rezolvării de probleme.	Evaluare cu notă în cadrul unei testări de laborator la sfârșitul semestrului, cu pondere de 75% din nota de laborator. Evaluare continuă cu notă a temelor de laborator + activității în timpul orelor de laborator, cu pondere de 25%.	25%
10.6 Standard minim de performanță			
Conținuturi: Competențe profesionale (R.Î.3.1.1) Absolventul definește conceptele următoare: grafuri orientate și neorientate, arbori binari de căutare, heap-uri binare, arbori roșu-negru (R.Î.3.1.2) și (R.Î.4.1.1). Absolventul compară și distinge structurile de bază precum vectori, matrice stive, cozi, liste, pe baza proprietăților și complexității operațiilor (R.Î.4.1.2.) Absolventul știe unde în problemele reale se folosesc structurile de date vectori, matrice, stive și cozi,. (R.Î.4.3.1). Absolventul poate să scrie cod pentru probleme simple ce folosesc vectori și matrice în limbajul C++ (R.Î. 5.1.) Absolventul poate să scrie un cod fără erori de compilare și de execuție pentru probleme simple ce folosesc vectori și matrice în limbajul C++ (R.Î. 5.3). Absolventul poate să descrie instrucțiunile de bază ale limbajului C++ precum și containerele de bază pentru structuri de date din biblioteca STL (respectiv: struct, vector, stack, queue) (R.Î. 5.4). Absolventul poate să explice funcționarea programelor implementate, precum și complexitatea de timp a acestora. (R.Î. 5.5). Absolventul poate să dezvolte programe care rezolvă probleme simple ce folosesc vectori și matrice în limbajul C++			

Competențe transversale

(RÎ 2.1): Absolventul știe să acceseze platforma de e-learning pentru a încărca la timp temele de laborator

(RÎ 2.3): Absolventul scrie și aplică instrucțiuni simple, în vederea rezolvării unor probleme sau a îndeplinirii unor sarcini de bază, în timpul orelor de laborator pentru rezolvarea fișelor de lucru.

Restricții:

Pentru promovarea examenului sunt necesare o notă de minim 5 la examenul scris și o medie de minim 5 între nota de la examenul scris și nota de la laborator.

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. Gabriel Stan	Director de departament Conf. dr. Nicușor Minculete
Titular Curs Lect. dr. Ioana Cristina Plajer	Titular de laborator Drd. Iulian Popa

Notă:

- 1) Domeniul de studii – se alege una din variantele: Licență / Masterat (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.
- 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 impusă) / DO (disciplină opțională) / DFac (disciplină facultativă).
- 5) Un credit este echivalent cu 30 de ore de studiu (activități didactice și studiu individual).