



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea VALAHIA din Târgoviște
1.2 Facultatea/Departamentul	Inginerie Electrică, Electronică și Tehnologia Informației
1.3 Departamentul	Electronică, Telecomunicații și Inginerie Energetică
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
1.5 Ciclul de studii	Licență
1.6 Programul de studii/Calificarea	Tehnologii și Sisteme de Telecomunicații

2. Date despre disciplină

2.1 Denumirea disciplinei	Structuri de date și algoritmi						
2.2 Titularul activităților de curs	Șl. univ. dr. ing. Lucia PASCALE						
2.3 Titularul activităților de seminar	As. univ. dr. Ionuț Adrian TUDOROIU						
2.4 Anul de studiu	I	2.5 Semestrul	II	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	B F

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

3.1 Număr de ore pe săptămână	3	din care: 3.2 curs	1	3.3 seminar/laborator	2L
3.4 Total ore din planul de învățământ	42	din care: 3.5 curs	14	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					26
Tutoriat					16
Examinări					2
Alte activități					0
3.7 Total ore studiu individual					83
3.9 Total ore pe semestru					125
3.10 Numărul de credite					5

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea Calculatoarelor și Limbaje de Programare
4.2 de competențe	Gândire logică, structurată. Cunoștințe de sintaxă a limbajelor C și C++

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	Sala cu videoproiector (pentru activități față în față) / Microsoft Teams și Moodle (pentru activități online)
5.2 de desfășurare a seminarului/laboratorului	Sala cu PC-uri (desktop sau laptop) (pentru activități față în față) / Microsoft Teams și Moodle (pentru activități online)

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

6.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">Dobândirea cunoștințelor necesare înțelegerii, adaptării și implementării de algoritmi de complexitate medie și ridicată, precum
---------------------------------------	--

	și alegerii algoritmului celui mai performant pentru rezolvarea unei probleme.
6.2 Obiectivele specifice	<ul style="list-style-type: none"> • Însușirea noțiunilor de analiză a problemei și de algoritmistică. • Capabilitatea de a analiza performanțele unui algoritm • Capabilitatea de a evalua complexitatea unei probleme

7. Rezultatele învățării

7.1 Cunoștințe (<i>Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau faptice</i>)
<ul style="list-style-type: none"> • Cunoștințe legate de sintaxa limbajului C++ și de modul de implementare a structurilor de date uzuale, precum și avantajele și dezavantajele utilizării lor în diverse scenarii.
7.2 Aptitudini (<i>Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente))</i>)
<ul style="list-style-type: none"> • Aptitudini legate de a analiza o problemă dată în limbaj natural și a dezvolta o soluție proprie pentru rezolvarea problemei, precum și capacitatea de a implementa această soluție în limbajul C++, folosind structurile de date uzuale și biblioteca STL.
7.3 Responsabilitate și autonomie (<i>Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale</i>)
<ul style="list-style-type: none"> • Capacitatea de a căuta răspunsuri și a se documenta online legat de sintaxa și API-ul limbajului C++, precum și de a căuta soluții la probleme punctuale.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
C.1. Recapitulare unor noțiuni din C++: funcții, structuri de date, pointeri. Clase de complexitate. Analiza complexității algoritmilor.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
C.2. Recursivitate directă și indirectă. Algoritmi reprezentativi. Tehnici de sortare. Tehnici simple de sortare: selecție, inserție, interschimbare.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
C.3. Tehnici avansate de proiectare. Algoritmi Divide et impera. Algoritmi Greedy.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
C.4. Tehnici avansate de proiectare. Algoritmi backtracking nerecursivi și recursivi.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
C.5. Programare dinamică. Noțiuni introductive. Structuri dinamice de date. Liste liniare simplu înlănțuite. Liste liniare dublu înlănțuite. Liste circulare. Stive. Cozi.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
C.6. Structuri de date. Arbori. Arbori binari de căutare. Refacerea arborilor binari. Forma poloneză a expresiilor aritmetice.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
C.7. Grafuri neorientate. Grafuri orientate.	explicația, prelegerea, conversația, problematizarea – expunere videoproiector	2 ore
Bibliografie 1. Pascale L.- Note de curs, Proiectarea algoritmilor, disponibile pe platforma http://moodle.valahia.ro/ 2. Oana-Carmen Niculescu-Faida, Tehnici de programare C++, Editura MatrixRom, București, 2023 3. https://visualstudio.microsoft.com/vs/getting-started/		
8.2 Seminar/laborator	Metode de predare	Observații
L.1. Recapitulare unor noțiuni din C++: funcții, structuri de date, pointeri. Clase de complexitate. Analiza complexității algoritmilor.	Problematizarea, Reflecția personală, Exercițiul, Dezbaterile, Studiul de caz	4 ore
L.2. Recursivitate directă și indirectă. Algoritmi reprezentativi. Tehnici de sortare. Tehnici simple de sortare: selecție, inserție, interschimbare.	Problematizarea, Reflecția personală, Exercițiul, Dezbaterile, Studiul de caz	4 ore
L.3. Tehnici avansate de proiectare. Algoritmi Divide et impera. Algoritmi Greedy.	Problematizarea, Reflecția personală, Exercițiul, Dezbaterile, Studiul de caz	4 ore
L.4. Tehnici avansate de proiectare. Algoritmi backtracking nerecursivi și recursivi.	Problematizarea, Reflecția personală, Exercițiul, Dezbaterile, Studiul de caz	4 ore
L.5. Programare dinamică. Noțiuni introductive. Structuri dinamice de date. Liste liniare simplu înlănțuite. Liste liniare dublu înlănțuite. Liste circulare. Stive. Cozi.	Problematizarea, Reflecția personală, Exercițiul, Dezbaterile, Studiul de caz	4 ore

L.6. Structuri de date. Arbori. Arbori binari de căutare. Refacerea arborilor binari. Forma poloneza a expresiilor aritmetice.	Problematicizarea, Reflecția personal, Exercițiul, Dezbateră, Studiul de caz	4 ore
L.7. Grafuri neorientate. Grafuri orientate.	Problematicizarea, Reflecția personal, Exercițiul, Dezbateră, Studiul de caz	4 ore

Bibliografie

1. Tudoroiu I. A. - Lucrări de laborator, Proiectarea algoritmilor, disponibile pe platforma <http://moodle.valahia.ro/>
2. Oana-Carmen Niculescu-Faida, Tehnici de programare C++, Editura MatrixRom, București, 2023
3. <https://visualstudio.microsoft.com/vs/getting-started/>

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

- Întreprinderi și firme de profil din zonele și județele limitrofe orașului Târgoviște.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	-răspunsurile la evaluarea finală - alte activități	- Examinare scrisă privitoare la însușirea cunoștințelor teoretice și aplicative	Evaluare finală (40%) - Test scris/grilă
10.5 Seminar/laborator	- răspunsurile finale la lucrarea de laborator - elaborarea și prezentarea temei de casă - răspunsuri la activitățile de laborator	- Monitorizarea periodică a progresului în cadrul ședințelor de laborator - Examinare practică lucrare de laborator	Laborator (30%) - Test scris/grilă Predarea și prezentarea temei de casă (30%)
10.6 Standard minim de performanță			
Pentru promovarea disciplinei, studentul trebuie să îndeplinească cumulativ următoarele condiții:			
<ul style="list-style-type: none"> - obține minimum 50% din punctajul total cumulat și minimum 50% din punctajul aferent fiecărei componente de evaluare (examen final, activitate de laborator și temă de casă); - demonstrează cunoștințe fundamentale (RI – Cunoștințe) privind structurile de date uzuale, tehnicile de proiectare a algoritmilor și analiza complexității acestora; - aplică aptitudini specifice (RI – Aptitudini) pentru proiectarea, implementarea și optimizarea algoritmilor utilizând limbajul C/C++ și structuri de date adecvate; - utilizează metode și instrumente (RI – Aptitudini) pentru analiza performanței algoritmilor și alegerea soluțiilor eficiente pentru probleme date; - manifestă responsabilitate și autonomie (RI – Responsabilitate și autonomie) în realizarea activităților de laborator, dezvoltarea temelor și documentarea soluțiilor; - argumentează soluțiile implementate, demonstrând capacitate de analiză, gândire algoritmică și optimizare a performanței. 			
Standardul minim de performanță validează atingerea rezultatelor învățării definite la punctul 7 (7.1–7.3).			

Fișa disciplinei corespunde planului de învățământ care se aplică pentru anul I începând cu anul universitar 2026-2027.

Data completării
07.09.2026

Titularul de curs
Șl. univ. dr. ing. Lucia PASCALE

Titularul de aplicații
As. univ. dr. Ionuț Adrian TUDOROIU

Data avizării în
departament
23.09.2026

Director de departament
Conf. univ. dr. ing. Dan Constantin PUCHIANU

Data avizării în
Consiliul Facultății
24.09.2026

Decan
Conf. univ. dr. ing. Nicoleta ANGELESCU