



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	TEORIA PROBABILITĂȚILOR ȘI STATISTICĂ MATEMATICĂ – TIDF201						
1.2 Titularul activităților de curs	Lect. Dr. Ion Adelina						
1.3 Titularul activităților de seminar	Lect. Dr. Ion Adelina						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DF

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	2/0
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	14/0
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Analiză Matematică I (TIDF101), Analiză Matematică II (TIDF109) — derivate, integrale, serii; Algebră Liniară (TIDF102) — matrice, sisteme liniare, spații vectoriale; Matematici Speciale (TIDF110) — ecuații diferențiale, transformate.
3.2 de competențe	Stăpânirea calculului diferențial și integral (derivate, integrale definite, serii); cunoașterea elementelor de algebră liniară (matrice, determinanți, sisteme liniare); capacitatea de a lucra cu sume, produse și limite; abilitatea de a manipula formal expresii matematice cu rigurozitate.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă (pentru demonstrații matematice), videoproiector și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea notelor de curs, culegerii de probleme și temelor; recomandare: instalarea Python (scipy, numpy, matplotlib) sau R pe laptopul propriu înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Sală de seminar cu tablă; studenții lucrează individual și în perechi la rezolvarea problemelor propuse; calculatorul sau telefonul mobil poate fi folosit pentru

verificarea calculelor cu Wolfram Alpha sau Python; problemele sunt afișate pe platforma Sakai cu o săptămână înainte de seminar.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C2. Posedă cunoștințe specializate de matematici discrete, teoria probabilităților și statistică matematică, necesare proiectării algoritmilor și analizei datelor în sisteme informatice complexe. C22. Posedă cunoștințe specializate privind principiile proiectării interfețelor grafice și metodologiile de proiectare centrată pe utilizator, cu capacitatea de a evalua critic uzabilitatea și accesibilitatea soluțiilor software.
5.2 Aptitudini	A7. Aplică metode numerice și tehnici de modelare matematică pentru simularea și rezolvarea problemelor ingineresti, utilizând instrumente software specializate de calcul numeric. A36. Aplică metode de estimare statistică și decizie bayesiană în analiza datelor experimentale și în proiectarea sistemelor de detecție, evaluând performanța estimatorilor și selectând criteriile optime de decizie.
5.3 Responsabilitate și autonomie	R36. Își asumă responsabilitatea pentru aplicarea riguroasă a metodelor statistice de estimare și decizie în analiza datelor experimentale, documentând transparent ipotezele statistice și comunicând onest limitele concluziilor formulate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în teoria probabilităților. Noțiunea de experiment aleator, spațiu de evenimente, eveniment. Operații cu evenimente: reuniune, intersecție, complementară. Definiția clasică, statistică și axiomatică (Kolmogorov) a probabilității. Proprietăți ale probabilității. Spații de probabilitate finite și numărabile.	2	Prelegere, demonstrații la tablă	
Tema 2: Probabilități condiționate și independența evenimentelor. Probabilitatea condiționată: definiție și proprietăți. Regula de multiplicare. Formula probabilității totale. Formula lui Bayes și aplicații (clasificatorul Naive Bayes, testarea diagnostică). Independența evenimentelor: două și n evenimente. Scheme clasice: Bernoulli.	2	Prelegere, demonstrații, exemple TI	
Tema 3: Variabile aleatoare discrete. Definiție și funcție de masă de probabilitate (pmf). Funcția de repartiție (cdf). Valoare așteptată (medie), varianță, deviație standard. Covarianță și corelație. Inegalitatea lui Markov și inegalitatea lui Cebîșev. Funcții generatoare de probabilitate.	2	Prelegere, demonstrații la tablă	
Tema 4: Repartiții discrete clasice. Repartiția Bernoulli, Binomială, Geometrică, Hipergeometrică, Poisson. Proprietăți și parametri. Relații între repartiții. Aproximarea Binomială prin Poisson. Aplicații în TI: modelarea erorilor de transmisie, numărul de request-uri pe server, numărul de defecte în cod.	2	Prelegere, exemple aplicative TI	

Tema 5: Variabile aleatoare continue. Funcție de densitate de probabilitate (pdf) și funcție de repartiție (cdf). Valoare așteptată, varianță pentru v.a. continue. Cuantile și mediană. Transformări ale variabilelor aleatoare. Simularea variabilelor aleatoare: metoda inversă.	2	Prelegere, demonstrații la tablă	
Tema 6: Repartiții continue clasice. Repartiția Uniformă, Exponențială (și proprietatea de lipsă de memorie), Normală (Gauss). Standardizarea: scorul Z. Repartiția Log-Normală, Chi-Pătrat, t-Student, Fisher-Snedecor. Tabele și calcule cu repartiția normală. Aplicații: timpii de răspuns ai serverelor, toleranțele de fabricație.	2	Prelegere, exemple, utilizare tabele	
Tema 7: Vectori aleatori și repartiții multidimensionale. Repartiția comună a doi vectori aleatori: pmf/pdf conjuncte, marginale, condiționate. Independența variabilelor aleatoare. Covarianță și matrice de covarianță. Repartiția normală bivariată. Suma variabilelor aleatoare independente.	2	Prelegere, demonstrații la tablă	
Tema 8: Legi ale numerelor mari și teorema limită centrală. Convergența în probabilitate și aproape sigură. Legea slabă și legea tare a numerelor mari. Teorema limită centrală (TLC): enunț și demonstrație pentru cazul i.i.d. Aproximarea normală a Binomialiei. Aplicații: eșantionarea, centrale limit theorem în statistică.	2	Prelegere, demonstrații, simulări Python	
Tema 9: Introducere în statistica matematică. Populație și eșantion. Statistici și estimatori. Proprietăți ale estimatorilor: nedepășare, consistență, eficiență. Metoda momentelor. Metoda verosimilității maxime (MLE): principiu, calcul pentru distribuții uzuale, proprietăți asimptotice.	2	Prelegere, calculul MLE la tablă	
Tema 10: Intervale de încredere. Intervalul de încredere pentru media unei populații normale cu varianță cunoscută (z-interval) și necunoscută (t-interval). Intervalul de încredere pentru proporție. Intervalul de încredere pentru varianță (Chi-Pătrat). Determinarea dimensiunii eșantionului. Interpretarea corectă a intervalelor de încredere.	2	Prelegere, exerciții numerice	
Tema 11: Testarea ipotezelor statistice. Ipoteza nulă și ipoteza alternativă. Erori de tip I și tip II. Puterea testului. Valoarea p (p-value): definiție, interpretare și limitele ei. Testul z, testul t (unilateral, bilateral, două eșantioane). Testul Chi-Pătrat pentru bunătața ajustării și pentru independență.	2	Prelegere, exerciții, interpretare p-value	
Tema 12: Regresia liniară simplă și multiplă. Modelul de regresie liniară simplă: estimarea coeficienților prin metoda celor mai mici	2	Prelegere, calculul OLS, Python/R	

pătrate (OLS). Testele pentru coeficienți (t-test). Coeficientul de determinare R^2 . Analiza reziduurilor. Regresia liniară multiplă: matrice de design, estimator OLS, colinearitate. Aplicații în Machine Learning.			
Tema 13: Introducere în statistică bayesiană. Probabilitate bayesiană vs. frecventistă. Teorema lui Bayes pentru densități: prior, likelihood, posterior. Exemple: inferența pentru proporție cu prior Beta, pentru medie cu prior normal. Estimatorul MAP (maximum a posteriori). Clasificatorul Naïve Bayes. Comparatie cu MLE.	2	Prelegere, exemple, aplicații NLP	
Tema 14: Procese stocastice — introducere și aplicații în TI. Definiție și tipuri de procese stocastice: cu timp discret/continuu, spațiu discret/continuu. Lanțuri Markov cu timp discret: matrice de tranziție, distribuție staționară, clasificarea stărilor. Procesul Poisson. Aplicații: modelarea cozilor de așteptare (M/M/1), rețele de distribuție, simularea traficului de rețea. Recapitulare generală.	2	Prelegere interactivă, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Blitzstein, J.K., Hwang, J., Introduction to Probability, 2nd ed., CRC Press / Chapman and Hall, 2019. Disponibil gratuit la: https://projects.iq.harvard.edu/stat110/home 2. DeGroot, M.H., Schervish, M.J., Probability and Statistics, 4th ed., Pearson, 2012 (retipărit 2021). 3. Stoleriu, I., Teoria Probabilităților și Statistică Matematică — Note de curs, Universitatea Al. I. Cuza, Iași, 2022. Disponibil la: http://www.math.uaic.ro/~stoleriu/PSM2022.pdf <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Bertsekas, D., Tsitsiklis, J., Introduction to Probability, 2nd ed., Athena Scientific, MIT, 2008. 5. Wackerly, D., Mendenhall, W., Scheaffer, R., Mathematical Statistics with Applications, 7th ed., Cengage Learning, 2008 (retipărit 2020). 6. Ross, S.M., Introduction to Probability Models, 12th ed., Academic Press / Elsevier, 2019. <p>Resurse software:</p> <ol style="list-style-type: none"> 7. Harvard STAT110 — cursul Introduction to Probability (Blitzstein), videoclipuri, probleme, soluții: https://projects.iq.harvard.edu/stat110/home 8. SciPy Statistics — documentație Python pentru distribuții de probabilitate și teste statistice: https://docs.scipy.org/doc/scipy/reference/stats.html 9. Wolfram MathWorld — Statistics and Probability: https://mathworld.wolfram.com/topics/ProbabilityandStatistics.html 		
6.2 Seminar	Nr. ore	Metode de predare	Observații
Tema S1: Calcul de probabilități — exerciții. Rezolvarea problemelor clasice de probabilitate: urne, cărți de joc, zaruri. Aplicarea formulei probabilității totale și a formulei lui Bayes la scenarii concrete (testarea diagnostică, filtre spam). Exerciții cu scheme Bernoulli și binomiale.	2	Rezolvare de probleme la tablă și individual	
Tema S2: Variabile aleatoare discrete — exerciții. Calculul funcției de masă, al valorii așteptate, varianței pentru v.a. discrete. Probleme cu repartiții Binomială, Poisson și Geometrică. Aplicații TI: modelarea numărului de pachete corupte dintr-o	2	Rezolvare de probleme, simulare Python/R	

transmisie, numărul de cereri simulate cu repartiție Poisson.			
Tema S3: Variabile aleatoare continue — exerciții. Calculul probabilităților cu densitățile Normal, Exponențial, Uniform. Standardizarea și utilizarea tabelii normale. Calculul cuantilelor. Aplicații: timpii de procesare a unui request, durata de viață a unui server, latența rețelelor.	2	Rezolvare de probleme, tabele, Python	
Tema S4: Estimarea statistică — exerciții. Calculul estimatorilor MLE pentru distribuțiile Binomială, Poisson, Normală, Exponențială. Calculul și interpretarea intervalelor de încredere de 90%, 95%, 99% pentru medie și proporție. Determinarea dimensiunii eșantionului pentru o eroare maximă admisă.	2	Rezolvare sistematică, interpretare numerică	
Tema S5: Testarea ipotezelor — exerciții. Aplicarea testului z și testului t (unilateral și bilateral) la date reale. Calculul valorii p și decizia statistică la nivel de semnificație dat. Testul Chi-Pătrat pentru independența a două variabile categoriale. Interpretarea corectă a rezultatelor și comunicarea onestă a limitelor.	2	Rezolvare de probleme, Python/SciPy	
Tema S6: Regresia liniară — exerciții. Calculul coeficienților OLS pentru un set de date mic (manual și Python). Trasarea dreptei de regresie. Calculul R^2 și interpretarea sa. Testul t pentru coeficienți. Analiza grafică a reziduurilor: detectarea violărilor ipotezelor modelului. Compararea a două modele de regresie.	2	Rezolvare de probleme, Python/scipy/sklearn	
Tema S7: Lanțuri Markov și procese stocastice — exerciții. Construirea matricei de tranziție pentru un sistem cu stări finite. Calculul distribuției după n pași. Determinarea distribuției staționare prin rezolvarea sistemului $p = pP$. Aplicație: modelarea unui server cu coadă M/M/1. Recapitulare generală și modele de subiecte de examen.	2	Rezolvare de probleme, recapitulare, simulare	

Bibliografie obligatorie seminar

1. Blitzstein, J.K., Hwang, J., Introduction to Probability, 2nd ed., CRC Press, 2019 — problemele din fiecare capitol (cu soluții disponibile online).
2. DeGroot, M.H., Schervish, M.J., Probability and Statistics, 4th ed., Pearson, 2012 — exercițiile de la finalul capitolelor.
3. Stoleriu, I., Teoria Probabilităților și Statistică Matematică — Note de curs, UAIC Iași, 2022: <http://www.math.uaic.ro/~stoleriu/PSM2022.pdf>

Bibliografie complementara seminar

4. Ross, S.M., Introduction to Probability Models, 12th ed., Academic Press, 2019 — exercițiile practice.
5. Wackerly, D., Mendenhall, W., Scheaffer, R., Mathematical Statistics with Applications, 7th ed., Cengage, 2008 — culegere de exerciții.

Resurse online gratuite recomandate

6. SciPy Stats — funcții Python pentru toate distribuțiile și testele statistice: <https://docs.scipy.org/doc/scipy/reference/stats.html>
7. StatKey (Lock5) — simulatoare interactive pentru distribuții, estimare și testare: <https://www.lock5stat.com/StatKey/>
8. Wolfram Alpha — verificarea calculelor de probabilitate și statistică: <https://www.wolframalpha.com/>

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

Teoria Probabilităților și Statistica Matematică furnizează fundamentul matematic al domeniilor cu cea mai rapidă creștere din TI: Machine Learning, Inteligența Artificială, securitatea cibernetică și analiza datelor. Algoritmii de clasificare (Naïve Bayes, logistic regression, SVM), modelele probabilistice generative (VAE, difuzion models), sistemele de detecție a anomaliilor și testele A/B din industria software sunt direct construite pe conceptele predate în această disciplină. Fără o înțelegere solidă a variabilelor aleatoare, distribuțiilor, estimării și testării de ipoteze, inginerul TI nu poate evalua critic rezultatele unui model ML sau nu poate interpreta corect un raport de analiză de date.

Conținuturile disciplinei sunt alinate curriculum-ului ACM/IEEE CS2023 pentru Computer Science, care include Probabilistic Analysis și Statistics ca elemente obligatorii ale formării inginerilor în calculatoare. Componenta de statistică bayesiană (Tema 13), introdusă explicit în această fișă, răspunde tendințelor actuale din industria TI: inferența bayesiană stă la baza algoritmilor de filtrare spam, a sistemelor de recomandare și a modelelor de limbaj de mari dimensiuni (LLM). Angajatorii din industria de Data Science și ML confirmă că probabilitățile și statistica sunt competențele matematice cel mai frecvent testate în interviurile tehnice.

Legătura cu disciplinele din anii următori este directă: Inteligența Artificială și Machine Learning (sem. 5-6) utilizează distribuții de probabilitate, estimatori și testare de ipoteze; Prelucrarea Semnalelor utilizează procese stocastice; Securitatea Informatică aplică teoria probabilităților în criptografie și în analiza atacurilor. Seminarul, cu accent pe rezolvarea de probleme și pe simularea în Python/R, urmărește formarea deprinderilor de calcul necesare înțelegerii critice — nu doar reproductive — a metodelor statistice folosite în instrumentele TI moderne.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea definițiilor și proprietăților fundamentale ale probabilității; stăpânirea principalelor distribuții discrete și continue (parametri, formule, aplicații); înțelegerea estimatorilor MLE și a proprietăților lor; capacitatea de a construi și interpreta intervale de încredere și de a efectua un test de ipoteze cu valoare p ; cunoașterea principiilor regresiei liniare și ale statisticii bayesiene.	Examen scris (3 ore): 4 probleme — câte una din: calcul de probabilități și distribuții, estimare statistică și intervale de încredere, testare de ipoteze și o problemă sinteză din regresie sau procese stocastice — cu demonstrarea completă a calculelor și justificarea ipotezelor utilizate	70%
8.5 Seminar/laborator	Corectitudinea și rigoarea rezolvărilor la seminar; capacitatea de a selecta metoda statistică adecvată unui context dat și de a justifica alegerea;	Evaluare continuă: participare activă la seminar și 3 teme de casă (câte una după S2, S4, S6) cu rezolvări complete și justificate	30%

	calitatea temelor de casă (calcul complete, ipoteze explicitate, concluzii formulate); participarea activă și progresul demonstrat pe parcursul semestrului.		
--	--	--	--

8.6 Standard minim de performanță:

Rezolvarea corectă și completă la examen a cel puțin două din cele patru probleme, cu obligativitatea de a include în mod necesar problema de estimare statistică (calculul estimatorului, intervalul de încredere și interpretarea lui în context) — aceasta verificând competența centrală a disciplinei. Simpla aplicare a formulelor fără justificarea ipotezelor și fără interpretarea rezultatelor nu constituie rezolvare completă și nu îndeplinește standardul minim.

Data completării 17.03.2026	Semnătura titularului de curs Lect. Dr. Ion Adelina	Semnătura titularului de seminar / laborator Lect. Dr. Ion Adelina
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		MĂSURĂTORI ELECTRONICE, SENZORI ȘI TRADUCTOARE – TIDS202					
1.2 Titularul activităților de curs		Conf. Dr. Aramă Iulian-Nicușor					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Conf. Dr. Aramă Iulian-Nicușor					
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					69
Documentare					20
Studiu individual					20
Referate					20
Teme casă					3
Proiect					2
Alte activități (inclusive colocviu, examen)					4
2.9 Total ore pe semestru					125
2.10 Numărul de credite					5

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Electrotehnică (TIDS113) — circuite electrice, teoria semnalelor; Dispozitive Electronice și Electronică Analogică (TIDS114) — amplificatoare, traductoare de semnal, circuite de condiționare; Fizică (TIDF103) — câmpuri fizice, fenomene fizice la baza principiilor de măsurare.
3.2 de competențe	Capacitatea de a analiza circuite electrice cu amplificatoare operaționale; cunoașterea principiilor de bază ale convertoarelor analog-digitale și digital-analogice; înțelegerea noțiunilor de semnal, zgomot, frecvență și bandă de trecere; abilitatea de a utiliza un multimetru și un osciloscop la nivel elementar.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, foilor de catalog ale senzorilor și temelor; recomandare: instalarea NI LabVIEW Student Edition sau Python (numpy, matplotlib, scipy) înainte de curs.
-------------------------------	--

4.2 de desfășurare a seminarului/laboratorului	Laborator de măsurări electronice dotat cu: surse de alimentare reglabile DC, generatoare de semnal, oscilosoape digitale, multimetre digitale de precizie, punți de măsurare, traductoare de temperatură (PT100, NTC, termocuplu), presiune, umiditate, accelerație (MEMS); plăci de achiziție date (NI USB-6000 sau echivalent Arduino/Raspberry Pi); calculatoare cu LabVIEW sau Python instalat.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C30. Cunoaște principiile achiziției și prelucrării datelor — conversie analog-digitală, eșantionare, filtrare numerică, senzori și traductoare — cu conștientizarea critică a limitelor de precizie și a surselor de eroare în sistemele de măsurare.
5.2 Aptitudini	A28. Configurează și utilizează sisteme de achiziție a datelor — plăci DAQ, interfețe senzori, lanțuri de măsurare — programând rutine de eșantionare, filtrare și vizualizare a datelor în timp real. A42. Analizează și interpretează date experimentale din sisteme de măsurare și senzori, aplicând tehnici statistice de prelucrare a semnalelor pentru extragerea informațiilor relevante și identificarea erorilor sistematice și aleatorii.
5.3 Responsabilitate și autonomie	R28. Gestionează autonom configurarea și mentenanța sistemelor de achiziție a datelor, asumându-și responsabilitatea pentru calibrarea periodică, documentarea lanțurilor de măsurare și validarea integrității datelor colectate. R42. Acționează responsabil în interpretarea și comunicarea rezultatelor măsurătorilor și experimentelor tehnice, documentând complet condițiile de măsurare, sursele de incertitudine și consecințele practice ale rezultatelor obținute.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în metrologie și măsurători electronice. Mărimi fizice și unități de măsură (sistemul SI). Clasificarea instrumentelor de măsurare. Caracteristici metrologice: domeniu de măsurare, rezoluție, precizie, acuratețe, sensibilitate, liniaritate, repetabilitate. Erori de măsurare: sistematice, aleatoare, grosolane. Propagarea erorilor. Incertitudinea de măsurare (GUM).	2	Prelegere interactivă, exemple numerice	
Tema 2: Semnale electrice de măsurare. Clasificarea semnalelor: analogice și digitale, periodice și aperiodice. Reprezentarea în domeniul timpului și al frecvenței (transformata Fourier). Spectrul semnalului și banda de frecvență. Zgomotul în sistemele de măsurare: zgomot termic (Johnson), zgomot de grenăilare (shot noise), raportul semnal/zgomot (SNR). Tehnici de reducere a zgomotului: blindare, torsadare, filtrare.	2	Prelegere interactivă, analiză spectrală	
Tema 3: Traductoare rezistive. Principiu de funcționare. Rezistențe dependente de temperatură: RTD (PT100, PT1000) — caracteristici, curba de calibrare, circuite de interfațare (punte Wheatstone cu 2, 3 și 4 fire). Termistori NTC și PTC: caracteristici,	2	Prelegere, demonstrații, foi de catalog	

aplicații. Traductoare de forță și deformație: rezistențe tensiometrice (strain gauge), factor de sensibilitate, amplificatoare de instrumentație.			
Tema 4: Traductoare capacitive și inductive. Traductoare capacitive: principiu, variante constructive, circuite de condiționare. Traductoare inductive: LVDT (Linear Variable Differential Transformer), principiu, caracteristici, aplicații în măsurarea deplasărilor. Inductanță mutuală și traductoare cu reluctanță variabilă. Traductoare de proximitate inductive. Aplicații industriale.	2	Prelegere interactivă, scheme	
Tema 5: Traductoare de temperatură active. Termocupluri: efectul Seebeck, tipuri standard (K, J, T, E, N, S, R, B), joncțiunea de referință, compensarea joncțiunii reci, circuite de condiționare, precizie și domeniu. Senzori de temperatură în circuit integrat: AD590, LM35, LM335, DS18B20 (1-Wire). Comparatie între tipurile de senzori de temperatură.	2	Prelegere interactivă, foi de catalog	
Tema 6: Traductoare de presiune și forță. Principii fizice: piezoresistiv, capacitiv, piezoelectric. Senzori de presiune MEMS: structură, funcționare, caracteristici. Celule de sarcină (load cells): tipuri, clase de precizie, instalare. Traductoare de presiune absolută, relativă și diferențială. Aplicații în sisteme embedded și IoT.	2	Prelegere interactivă, studii de caz	
Tema 7: Senzori de mișcare și poziție. Encodere incrementale și absolute: principiu optic și magnetic, rezoluție, interfețe (quadrature, SSI, BiSS). Senzori Hall: principiu, aplicații în detecția poziției și a curentului. Accelerometre și giroscopie MEMS: principiu, parametri (sensibilitate, offset, zgomot), interfețe (I2C, SPI). IMU (Inertial Measurement Unit). Aplicații în robotică și IoT.	2	Prelegere interactivă, foi de catalog	
Tema 8: Senzori de umiditate, luminozitate și gaze. Senzori capacitivi de umiditate relativă: principiu și circuite de condiționare. Senzori de luminozitate: fotorezistori, fotodiode, fototransistori, senzori de culoare RGB. Senzori de gaze: electrochimici, metal-oxid (MOS), NDIR. Senzori pentru calitatea aerului (VOC, CO, CO ₂). Aplicații în sisteme smart-home și mediu.	2	Prelegere interactivă, demonstrații	
Tema 9: Condiționarea semnalelor de la traductoare. Amplificatoare de instrumentație: principiu, CMRR, câștig, exemple (INA128, AD620). Filtre analogice: Butterworth, Chebyshev, Bessel — ordinul 1 și 2. Circuite de liniarizare. Izolarea galvanică a semnalelor (optocupluri, izolatori). Protecția la supratensiune a intrărilor analogice.	2	Prelegere interactivă, scheme	

<p>Tema 10: Conversia analog-digitală. Teorema eșantionării (Nyquist-Shannon). Filtrare anti-aliasing. Parametrii ADC: rezoluție, LSB, ENOB, SNR, THD+N, SFDR, offset, câștig, INL, DNL. Tipuri de ADC: succesive aproximații (SAR), Sigma-Delta, Flash, cu integrare. Circuite Sample-and-Hold. Alegerea ADC-ului în funcție de aplicație.</p>	2	Prelegere interactivă, calcule numerice	
<p>Tema 11: Sisteme de achiziție date (DAQ). Arhitectura unui sistem DAQ: multiplexor analogic, amplificator de câștig programabil (PGA), ADC, buffer FIFO, interfață digitală. Plăci DAQ tipice (NI USB-6001, LabJack). Rate de eșantionare, număr de canale, sincronizare. Achiziția simultană vs. secvențială. Triggering. Comunicarea cu PC-ul: USB, Ethernet, PCIe.</p>	2	Prelegere interactivă, demonstrații hardware	
<p>Tema 12: Programarea sistemelor DAQ — LabVIEW și Python. Paradigma programării în LabVIEW: Virtual Instruments (VI), dataflow. Achiziția de date în LabVIEW: DAQmx, citirea senzorilor analogici și digitali, generarea de semnale. Achiziția de date în Python: biblioteca nidaqmx, matplotlib pentru vizualizare în timp real. Comparatie LabVIEW vs. Python pentru aplicații DAQ.</p>	2	Prelegere interactivă, demonstrații software	
<p>Tema 13: Prelucrarea statistică a datelor de măsurare. Distribuții statistice: normală, t-Student, Chi-pătrat. Estimarea valorii medii și a incertitudinii standard. Testul Grubbs pentru identificarea valorilor aberante (outlieri). Regresia liniară și polinomială pentru calibrarea senzorilor. Intervalele de încredere pentru rezultatele măsurătorilor. Documentarea conform standardului ISO/IEC 17025.</p>	2	Prelegere interactivă, exerciții Python	
<p>Tema 14: Rețele de senzori și IoT. Protocoale de comunicare pentru senzori: I2C, SPI, UART, 1-Wire, CAN. Protocoale wireless: Bluetooth LE, Zigbee, LoRa, NB-IoT. Platforme IoT: MQTT, Node-RED, ThingSpeak, Home Assistant. Securitatea rețelelor de senzori. Integrarea senzorilor în sisteme embedded (Arduino, Raspberry Pi, ESP32). Recapitulare generală.</p>	2	Prelegere interactivă, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Fraden, J., Handbook of Modern Sensors: Physics, Designs, and Applications, 5th ed., Springer, 2016. 2. Webster, J.G. (ed.), Medical Instrumentation: Application and Design, 4th ed., Wiley, 2010 (retipărit 2019) — capitolele de traductoare și achiziție de date. 3. National Instruments, NI DAQmx Help and LabVIEW Documentation. Disponibil la: https://www.ni.com/docs/ <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Placko, D. (ed.), Fundamentals of Instrumentation and Measurement, ISTE / Wiley, 2007. 5. Pallás-Areny, R., Webster, J.G., Sensors and Signal Conditioning, 2nd ed., Wiley, 2001 (retipărit 2019). 		

	<p>6. Joint Committee for Guides in Metrology (JCGM), GUM: Guide to the Expression of Uncertainty in Measurement, BIPM, 2008. Disponibil la: https://www.bipm.org/en/committees/jc/jcgm/publications</p> <p>Resurse software:</p> <p>7. National Instruments — tutoriale LabVIEW și DAQmx: https://www.ni.com/en/support/documentation.html</p> <p>8. Adafruit Learning System — ghiduri pentru senzori și Arduino/Raspberry Pi: https://learn.adafruit.com/</p> <p>9. SparkFun Learn — tutoriale pentru senzori și sisteme embedded: https://learn.sparkfun.com/</p>		
6.2 Laborator	Nr. ore	Metode de predare	Observații
<p>Lucrarea L1: Familiarizarea cu echipamentele de laborator și norme de securitate. Utilizarea corectă a multimetrului digital (tensiune, curent, rezistență, continuitate, diodă). Utilizarea osciloscopului: calibrarea, setarea timebase și volts/div, măsurarea amplitudinii, frecvenței și fazei unui semnal sinusoidal. Măsurarea tensiunii și curentului în circuite simple.</p>	2	Lucrare practică individuală	
<p>Lucrarea L2: Erori de măsurare și incertitudinea de măsurare. Măsurarea repetată a aceleași mărimi (tensiunea unui regulator). Calculul mediei, deviației standard, incertitudinii standard și extinse ($k=2$). Identificarea surselor de eroare sistematică. Compararea a două instrumente cu precizie diferită. Redactarea unui raport de incertitudine conform GUM.</p>	2	Lucrare practică, calcul numeric	
<p>Lucrarea L3: Traductoare rezistive de temperatură — PT100. Montarea unui PT100 în circuit (punte Wheatstone cu 3 fire). Măsurarea rezistenței la diferite temperaturi (gheață, apă, saună). Trasarea curbei de calibrare $R(T)$. Calculul erorii față de caracteristica standard IEC 60751. Compararea cu un termometru etalon.</p>	2	Lucrare practică, calibrare	
<p>Lucrarea L4: Termistori NTC și termocupluri. Caracterizarea unui termistor NTC: trasarea curbei $R(T)$ și calculul constantei B. Conectarea unui termocuplu de tip K cu compensarea joncțiunii reci (circuit cu LM35). Compararea PT100, NTC și termocuplu la aceeași temperatură. Analiza zgomotului în fiecare tip de senzor.</p>	2	Lucrare practică, comparație senzori	
<p>Lucrarea L5: Senzori de temperatură în circuit integrat. Caracterizarea LM35 (tensiune de ieșire vs. temperatură). Interfațarea DS18B20 (1-Wire) cu Arduino: citirea temperaturii prin protocol 1-Wire, afișarea pe LCD. Compararea LM35, DS18B20 și PT100 ca precizie, cost și ușurință de interfațare.</p>	2	Lucrare practică, Arduino	
<p>Lucrarea L6: Celule de sarcină și strain gauge. Montarea unui strain gauge pe o bară elastică și conectarea în punte Wheatstone completă (4 elemente). Amplificarea semnalului cu un amplificator de instrumentație (INA128). Calibrarea sistemului cu greutăți cunoscute.</p>	2	Lucrare practică, calibrare	

Trasarea curbei de calibrare forță-tensiune și calculul sensibilității.			
Lucrarea L7: Senzori de presiune și accelerație MEMS. Caracterizarea unui senzor de presiune MEMS (MPX5700 sau similar): răspuns la variația presiunii cu o pompă manuală. Interfațarea unui accelerometru MEMS (MPU-6050, I2C) cu Arduino: citirea accelerației pe 3 axe, detectarea orientării și a vibrațiilor.	2	Lucrare practică, MEMS, I2C	
Lucrarea L8: Achiziția de date cu placă DAQ — LabVIEW. Configurarea unui canal de achiziție analogic pe placa DAQ (NI USB-6001 sau LabJack). Crearea unui VI LabVIEW pentru achiziția și vizualizarea în timp real a semnalului unui senzor de temperatură. Setarea ratei de eșantionare și a numărului de eșantioane. Salvarea datelor în fișier CSV.	2	Lucrare practică LabVIEW / Python	
Lucrarea L9: Filtrare numerică a semnalelor achiziționate. Achiziția unui semnal cu zgomot (senzor + interferențe 50 Hz). Implementarea în Python a unui filtru trece-jos FIR și IIR (Butterworth, ord. 2). Vizualizarea comparativă a semnalelor filtrat și nefiltrat în domeniul timp și frecvență (FFT). Alegerea frecvenței de tăiere optime.	2	Lucrare practică Python/scipy	
Lucrarea L10: Calibrarea senzorilor și regresia liniară. Achiziția datelor de calibrare pentru un senzor de umiditate capacitiv (DHT22). Aplicarea regresiei liniare în Python (numpy.polyfit) pentru a obține curba de calibrare. Calculul erorii reziduale și a intervalului de încredere 95%. Compararea cu calibrarea din datasheet.	2	Lucrare practică Python, regresie	
Lucrarea L11: Rețea de senzori cu Arduino și transmisie serială. Conectarea a 3 senzori diferiți (temperatură, umiditate, lumină) la Arduino Mega. Programarea Arduino pentru citirea periodică a senzorilor și transmiterea datelor prin UART (Serial). Recepția și vizualizarea datelor pe PC cu Python (pyserial, matplotlib).	2	Lucrare practică Arduino + Python	
Lucrarea L12: Achiziție și vizualizare IoT cu ESP32 și MQTT. Conectarea unui senzor de temperatură/umiditate (DHT22) la ESP32. Programarea ESP32 pentru publicarea datelor pe un broker MQTT (local sau cloud). Subscrierea la topicul MQTT cu Python și vizualizarea datelor în timp real. Configurarea unui dashboard Node-RED.	2	Lucrare practică ESP32, IoT	
Lucrarea L13: Detectarea valorilor aberante și prelucrarea statistică. Achiziția unui set de 200 de măsurători ale aceluiași senzor. Aplicarea testului Grubbs pentru identificarea outlier-ilor. Calculul histogramei și verificarea	2	Lucrare practică Python, statistică	

normalității (testul Shapiro-Wilk cu scipy). Calculul incertitudinii de tip A și tip B. Redactarea unui raport de măsurare conform ISO/IEC 17025.			
Lucrarea L14: Proiect integrat de sistem de monitorizare și colocviu. Proiectarea și implementarea unui sistem complet de monitorizare: cel puțin 2 senzori eterogeni + DAQ + filtrare + calibrare + vizualizare în timp real + salvare CSV. Prezentarea orală a soluției (10 minute): senzori aleși, schema de conexiuni, codul, calibrarea, incertitudinea estimată. Colocviu pe baza întrebărilor din tematica disciplinei.	2	Proiect individual, colocviu oral	
<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Fraden, J., Handbook of Modern Sensors: Physics, Designs, and Applications, 5th ed., Springer, 2016 — capitolele de aplicații practice. 2. National Instruments, NI DAQmx Getting Started Guide și LabVIEW Core 1 Course Notes: https://www.ni.com/docs/ 3. Adafruit Learning System — îndrumătoare practice pentru senzori, Arduino și Raspberry Pi: https://learn.adafruit.com/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Pallás-Areny, R., Webster, J.G., Sensors and Signal Conditioning, 2nd ed., Wiley, 2001 — secțiunile de interfațare practică. 5. SciPy Signal Processing Documentation — filtrare numerică, FFT, statistică: https://docs.scipy.org/doc/scipy/reference/signal.html 6. Joint Committee for Guides in Metrology, GUM: Guide to the Expression of Uncertainty in Measurement, BIPM, 2008: https://www.bipm.org/en/committees/jc/jcgm/publications <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 7. SparkFun Learn — tutoriale și scheme pentru senzori și sisteme embedded: https://learn.sparkfun.com/ 8. Arduino Reference și tutoriale oficiale: https://www.arduino.cc/reference/en/ 9. SciPy Documentation — statistică și prelucrare semnale în Python: https://docs.scipy.org/doc/scipy/reference/signal.html 10. Random Nerd Tutorials — ESP32, MQTT, Node-RED, senzori IoT: https://randomnerdtutorials.com/ 			

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

Disciplina Măsurători Electronice, Senzori și Traductoare constituie puntea esențială dintre lumea fizică și sistemele informatice: orice sistem embedded, IoT, robotic sau de control nu poate funcționa fără achiziția corectă, condiționarea și interpretarea semnalelor provenite din mediu fizic. Înțelegerea principiilor de funcționare ale senzorilor, a lanțurilor de măsurare și a limitelor de precizie ale sistemelor DAQ este indispensabilă inginerului TI care proiectează sisteme cyber-fizice, platforme IoT sau aplicații de Machine Learning pe date de senzori.

Conținuturile sunt aliniate standardelor internaționale din domeniu — GUM (JCGM 100:2008) pentru evaluarea incertitudinii de măsurare și ISO/IEC 17025 pentru documentarea rezultatelor — și acoperă tehnologiile MEMS de ultimă generație utilizate masiv în smartphone-uri, drone, vehicule autonome și echipamente medicale. Componenta de programare LabVIEW și Python (lucrările L8-L13) pregătește studentul pentru utilizarea platformelor industriale NI și pentru integrarea senzorilor în sisteme software reale, competențe direct solicitate de companiile din domeniul embedded, automotive și industrial IoT.

Proiectul integrat din lucrarea L14 reproduce un flux de lucru real de inginerie: alegerea senzorilor, montajul, calibrarea, filtrarea, analiza statistică a incertitudinii și documentarea conform standardelor — toate condensate într-un sistem funcțional prezentat oral. Această abordare formează atât competența tehnică, cât și responsabilitatea inginerului față de corectitudinea datelor furnizate sistemelor informatice, critică în aplicații de siguranță (medical, automotive, industrial).

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea principiilor metrologice (erori, incertitudine, calibrare); înțelegerea funcționării traductoarelor rezistive, capacitive, inductive, termice și MEMS; cunoașterea parametrilor ADC și a arhitecturii sistemelor DAQ; capacitatea de a selecta senzorul adecvat unui context dat și de a estima incertitudinea de măsurare.	Examen scris (2 ore): 1 subiect teoretic de sinteză (principiul unui traductor și lanțul de măsurare), 2 probleme de calcul (erori/incertitudine + alegerea ADC pe baza specificațiilor) și 1 problemă de analiză a unui sistem DAQ complet — cu bareme detaliate	50%
8.5 Seminar/laborator	Corectitudinea și rigurozitatea măsurărilor și a calculelor de calibrare/incertitudine din rapoartele de laborator; calitatea proiectului integrat din L14 (funcționalitate, documentare, calibrare); nota la colocviul oral din L14 (explicarea soluției și răspunsuri la întrebări tehnice); participarea activă la toate lucrările.	Evaluare continuă: rapoarte de laborator individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la lucrarea L14 — sistem complet funcțional cu documentare și prezentare de 10 minute (50%)	50%
8.6 Standard minim de performanță:			
Demonstrarea funcțională în direct la colocviul oral din L14 a unui sistem de achiziție date complet — cu cel puțin doi senzori eterogeni conectați, semnal achiziționat, filtrat numeric și vizualizat în timp real — ȘI prezentarea documentației de calibrare cu incertitudinea estimată pentru cel puțin un senzor. Simpla afișare a unui cod fără demonstrarea funcționării live nu îndeplinește standardul minim. Predarea a cel puțin 10 din 14 rapoarte de laborator constituie condiție de acces la examen.			

Data completării 17.03.2026	Semnătura titularului de curs Conf. Dr. Aramă Iulian- Nicușor	Semnătura titularului de seminar / laborator Conf. Dr. Aramă Iulian- Nicușor
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	STRUCTURI DE DATE ȘI ALGORITMI – TIDS203						
1.2 Titularul activităților de curs	Lect. Dr. Lițan Daniela-Elena						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Lect. Dr. Lițan Daniela-Elena						
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					69
Documentare					20
Studiu individual					21
Referate					20
Teme casă					3
Proiect					2
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					125
2.10 Numărul de credite					5

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Programarea Calculatoarelor și Limbaje de Programare I (TIDF105) — algoritmi elementari, tipuri de date, tablouri, funcții; Programarea Calculatoarelor și Limbaje de Programare II (TIDF112) — programare structurată avansată, fișiere, pointeri; Matematici Discrete (TIDF201/semestrul următor) — grafuri, logică, combinatorică.
3.2 de competențe	Capacitatea de a scrie programe corecte în C/C++ sau Python (funcții, tablouri, pointeri/referințe, alocare dinamică de memorie); înțelegerea noțiunilor de bază din matematica discreta: mulțimi, relații, funcții; cunoașterea structurilor de control (if, for, while, recursivitate elementară).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoprojector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, pseudocodurilor și temelor; recomandare: instalarea unui IDE (Visual Studio Code + compilator GCC sau PyCharm) și a tool-ului de vizualizare algoritmi (VisuAlgo) înainte de curs.
-------------------------------	--

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale conectate la internet; IDE instalat: VS Code cu extensii C/C++ și Python, sau CLion/PyCharm; compilator GCC/G++ și Python 3.10+; acces la platforma de judecată automată (online judge) pentru testarea soluțiilor; studenții pot folosi propriul laptop.
--	--

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C8. Cunoaște în profunzime structurile de date și algoritmi fundamentali, inclusiv algoritmi paraleli și distribuiți, cu capacitatea de a analiza critic complexitatea și eficiența soluțiilor propuse.
5.2 Aptitudini	A6. Proiectează și analizează algoritmi eficienți pentru rezolvarea problemelor computaționale complexe, evaluând critic complexitatea temporală și spațială a soluțiilor propuse și alegând structurile de date adecvate.
5.3 Responsabilitate și autonomie	R8. Răspunde în mod responsabil și transparent față de echipă și beneficiari pentru respectarea termenelor, a bugetului tehnic și a specificațiilor agreeate, comunicând proactiv orice deviere sau risc identificat.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în analiza algoritmilor. Noțiunea de algoritm: corectitudine, terminare, eficiență. Analiza complexității: notații asimptotice O , Ω , Θ . Complexitate timp și spațiu. Analiza cazului cel mai rău, mediu, cel mai bun. Exemple: căutare liniară vs. binară. Recapitulare tipuri de date abstracte (ADT) și structuri de date elementare.	2	Prelegere interactivă, analiză exemple	
Tema 2: Tablouri, liste înlănțuite și structuri conexe. Tablouri unidimensionale și multidimensionale: acces, inserare, ștergere — complexitate. Liste simplu înlănțuite: nod, inserare la cap/coadă/poziție, ștergere, căutare. Liste dublu înlănțuite. Liste circulare. Comparatie tablou vs. listă înlănțuită: complexitate operații și consum memorie. Implementare în C/Python.	2	Prelegere, pseudocod, analiză complexitate	
Tema 3: Stive și cozi. Stiva (Stack): definiție ADT, implementare cu tablou și cu listă înlănțuită, operații push/pop/top. Coadă (Queue): definiție ADT, implementare cu tablou circular și cu listă înlănțuită, operații enqueue/dequeue/front. Coadă cu două capete (Deque). Aplicații: evaluarea expresiilor aritmetice (notație poloneză inversă), parcurgerea grafurilor.	2	Prelegere, demonstrații cod	
Tema 4: Arbori binari. Definiții: arbore, arbore binar, arbore binar complet, arbore binar perfect. Reprezentare prin structuri înlănțuite. Parcurgeri: preordine, inordine, postordine (recursiv și iterativ cu stivă), parcurgere pe niveluri (BFS). Numărul de noduri, înălțimea arborelui. Construirea unui arbore binar din șirurile de parcurgere.	2	Prelegere interactivă, vizualizări	

<p>Tema 5: Arbori binari de căutare (BST). Proprietatea BST. Operații: căutare, inserare, ștergere (cazurile unui copil, doi copii — succesorul inordine). Complexitate: caz mediu $O(\log n)$, caz defavorabil $O(n)$. Arbore BST degenerat. Introducere în arbori echilibrați: motivație AVL și arbori roșu-negru. Aplicații: dicționare, tabele de simboluri.</p>	2	Prelegere, analiză complexitate	
<p>Tema 6: Heap și coadă de priorități. Heap binar (min-heap și max-heap): proprietatea heap, reprezentare în tablou. Operații: insert (sift-up), extract-min/max (sift-down), buildHeap. Complexitate: $O(\log n)$ insert/extract, $O(n)$ buildHeap. Priority Queue ca ADT. Heapsort: algoritm și complexitate $O(n \log n)$. Heap-uri d-are. Aplicații: schedulare, algoritmi de drum cel mai scurt.</p>	2	Prelegere, demonstrații, analiză	
<p>Tema 7: Tabele de dispersie (Hash Tables). Funcții hash: proprietăți, funcții hash universale, funcții hash criptografice (SHA, MD5 — scurtă prezentare). Rezolvarea coliziunilor: înlănțuire (chaining) și adresare deschisă (sondare liniară, pătratică, dublă). Factor de încărcare. Complexitate: $O(1)$ mediu, $O(n)$ caz defavorabil. Aplicații: dicționare, cache, detectarea duplicatelor.</p>	2	Prelegere interactivă, analiză	
<p>Tema 8: Grafuri — reprezentare și parcurgere. Definiții: graf neorientat/orientat, ponderat, gradul unui nod. Reprezentare: matrice de adiacență, liste de adiacență — comparație spațiu și timp. Parcurgerea în adâncime (DFS): algoritm recursiv și iterativ, arborele DFS, clasificarea muchiilor. Parcurgerea în lățime (BFS): algoritm, arborele BFS. Aplicații: componente conexe, detectarea ciclurilor.</p>	2	Prelegere, vizualizări VisuAlgo	
<p>Tema 9: Algoritmi pe grafuri — drumuri și componente. Sortarea topologică (DFS + timestamps, Kahn cu coadă). Componente tare conexe: algoritmul Kosaraju, algoritmul Tarjan. Drumul cel mai scurt în grafuri fără ponderi (BFS). Algoritmul Dijkstra: pseudocod, demonstrație corectitudine, complexitate $O((V+E) \log V)$ cu heap. Algoritmul Bellman-Ford: grafuri cu ponderi negative, detectarea ciclurilor negative.</p>	2	Prelegere, pseudocod, analiză	
<p>Tema 10: Arbori de acoperire minimă și tehnici greedy. Arbore de acoperire minimă (MST): definiție, proprietatea tăieturii. Algoritmul Kruskal (union-find) și algoritmul Prim. Complexitate comparată. Tehnica greedy: principiu, corectitudine prin exchange argument. Alte aplicații greedy: problema fracționară a rucsacului, codificarea Huffman — construcția arborelui și codarea.</p>	2	Prelegere, demonstrații, analiză	
<p>Tema 11: Programare dinamică. Principiu: subprobleme suprapuse și substructură</p>	2	Prelegere interactivă, rezolvare pas cu pas	

<p>optimă. Abordări top-down (memoizare) vs. bottom-up (tabulare). Probleme clasice: Fibonacci cu memoizare, longest common subsequence (LCS), edit distance (distanța Levenshtein), problema rucsacului 0/1, cel mai lung subșir crescător (LIS). Complexitate timp și spațiu. Reconstrucția soluției.</p>			
<p>Tema 12: Divide-and-conquer și backtracking. Divide-and-conquer: paradigmă, teorema master pentru recurențe ($T(n) = aT(n/b) + f(n)$). Algoritmi: mergesort, quicksort (alegerea pivotului, quicksort randomizat), căutare binară, înmulțirea Strassen. Backtracking: paradigmă, spațiul stărilor, pruning. Probleme clasice: N-queens, Sudoku solver, suma submulțimilor, colorarea grafurilor.</p>	2	Prelegere, pseudocod, analiză recursivă	
<p>Tema 13: Algoritmi de sortare — analiză comparativă. Sortări prin comparație: insertion sort, selection sort, bubble sort — $O(n^2)$. Mergesort, heapsort — $O(n \log n)$. Quicksort — $O(n \log n)$ mediu, $O(n^2)$ worst case. Limita inferioară $\Omega(n \log n)$ pentru sortările prin comparație (demonstrație prin arbore de decizie). Sortări liniare: counting sort, radix sort, bucket sort — condiții și complexitate. Sortare stabilă vs. instabilă.</p>	2	Prelegere, comparație complexități	
<p>Tema 14: Structuri de date avansate și recapitulare. Arbori B și B+: motivație (I/O eficient), proprietăți, aplicații în baze de date. Skip lists: structură probabilistică, complexitate. Trie (arbore de prefix): operații insert/search/delete, aplicații în autocomplete și dicționare. Union-Find (Disjoint Set Union): union by rank, path compression — complexitate aproape $O(1)$. Recapitulare generală și probleme model de examen.</p>	2	Prelegere, recapitulare, probleme	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., Introduction to Algorithms (CLRS), 4th ed., MIT Press, 2022. 2. Sedgwick, R., Wayne, K., Algorithms, 4th ed., Addison-Wesley, 2011 (retipărit 2022). Disponibil gratuit la: https://algs4.cs.princeton.edu/ 3. Knuth, D.E., The Art of Computer Programming, Vol. 1–3, 3rd ed., Addison-Wesley, 2011. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Skiena, S.S., The Algorithm Design Manual, 3rd ed., Springer, 2020. 5. Goodrich, M.T., Tamassia, R., Goldwasser, M.H., Data Structures and Algorithms in Python, Wiley, 2013 (retipărit 2021). 6. Roughgarden, T., Algorithms Illuminated (Part 1–4), Soundlikeyourself Publishing, 2017–2020. Disponibil la: https://www.algorithmsilluminated.org/ <p>Resurse software:</p> <ol style="list-style-type: none"> 7. VisuAlgo — vizualizări interactive pentru structuri de date și algoritmi: https://visualgo.net/ 8. Princeton Algorithms Part I and II (Coursera/Sedgwick) — materiale video și exerciții: https://algs4.cs.princeton.edu/ 9. CP-Algorithms — referință completă de algoritmi competitivi cu implementări: https://cp-algorithms.com/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații

<p>Lucrarea L1: Analiza complexității algoritmilor — experimente empirice. Implementarea algoritmilor de căutare liniară și binară în Python/C++. Măsurarea timpului de execuție pentru $n \in \{1K, 10K, 100K, 1M\}$ cu biblioteca time. Compararea timpilor măsurați cu predicțiile teoretice $O(n)$ vs. $O(\log n)$. Reprezentarea grafică cu matplotlib.</p>	2	Lucrare practică individuală	
<p>Lucrarea L2: Liste înlănțuite — implementare de la zero. Implementarea în C++ (cu pointeri) sau Python a unei liste dublu înlănțuite cu operații complete: inserare la cap/coadă/ poziție, ștergere, căutare, inversare. Testarea cu suite de teste. Analiza memoriei alocate cu valgrind sau tracemalloc.</p>	2	Lucrare practică individuală	
<p>Lucrarea L3: Stive și cozi — aplicații. Implementarea unui evaluator de expresii aritmetice în notație poloneză inversă (RPN) cu stivă. Implementarea unui sistem de gestiune a task-urilor cu coadă de priorități simplă. Testarea cu exemple de intrare variate. Analiza comportamentului la overflow/underflow.</p>	2	Lucrare practică individuală	
<p>Lucrarea L4: Arbori binari — parcurgeri și operații. Implementarea unui arbore binar cu operații: inserare, ștergere, toate cele 4 parcurgeri (preordine, inordine, postordine, BFS). Verificarea proprietăților: înălțime, număr de noduri, echilibru. Reconstrucția unui arbore binar din parcurgerea preordine și inordine.</p>	2	Lucrare practică individuală	
<p>Lucrarea L5: BST, AVL — implementare și comparație. Implementarea unui BST complet (inserare, ștergere, căutare) și măsurarea înălțimii la inserare secvențială vs. aleatoare. Compararea timpilor de căutare BST vs. <code>std::map</code> (arbore roșu-negru din STL). Vizualizarea arborelui generat în funcție de ordinea inserării.</p>	2	Lucrare practică, benchmark	
<p>Lucrarea L6: Heap și heapsort. Implementarea unui min-heap de la zero cu operații insert, extract-min, buildHeap. Implementarea heapsort și verificarea corectitudinii. Benchmark: heapsort vs. mergesort vs. <code>std::sort</code> pe aceleași date. Implementarea unei cozi de priorități pentru simularea unui scheduler simplu.</p>	2	Lucrare practică, benchmark	
<p>Lucrarea L7: Tabele de dispersie — implementare și analiză coliziuni. Implementarea unui hash table cu chaining (liste înlănțuite la fiecare bucket). Testarea cu funcții hash diferite (modulo, multiplicare, FNV-1a). Analiza distribuției coliziunilor cu matplotlib. Implementarea redimensionării (rehashing) la depășirea factorului de încărcare.</p>	2	Lucrare practică individuală	

<p>Lucrarea L8: DFS și BFS pe grafuri. Implementarea grafurilor cu matrice de adiacență și liste de adiacență. Implementarea DFS recursiv și iterativ, BFS iterativ. Aplicații: detectarea ciclurilor, componente conexe, distanța minimă (fără ponderi). Testarea pe grafuri generate aleator și pe exemple clasice (rețele sociale, hărți).</p>	2	Lucrare practică individuală	
<p>Lucrarea L9: Algoritmul Dijkstra și sortare topologică. Implementarea algoritmului Dijkstra cu priority queue (heapq în Python sau priority_queue în C++). Testarea pe un graf ponderat reprezentând o rețea de rutere. Implementarea sortării topologice (algoritmul lui Kahn). Verificarea pe un DAG ce reprezintă dependențe între module software.</p>	2	Lucrare practică individuală	
<p>Lucrarea L10: Algoritmul Kruskal și Union-Find. Implementarea structurii Union-Find cu union by rank și path compression. Implementarea algoritmului Kruskal pentru MST. Testarea pe grafuri ponderate de dimensiuni variate. Benchmark față de algoritmul Prim (din librărie). Vizualizarea MST generat.</p>	2	Lucrare practică, benchmark	
<p>Lucrarea L11: Programare dinamică — probleme clasice. Implementarea soluțiilor DP pentru: LCS (longest common subsequence), edit distance (distanța Levenshtein) și problema rucsacului 0/1 — comparând abordarea top-down (memoizare) și bottom-up (tabulare). Reconstrucția soluției din matricea DP. Analiza complexității timp și spațiu.</p>	2	Lucrare practică individuală	
<p>Lucrarea L12: Divide-and-conquer — quicksort și mergesort. Implementarea mergesort și quicksort (cu pivot aleator). Benchmark comparat pe date sortate, invers sortate, aleatoare și cu duplicate — pentru $n \in \{10K, 100K, 1M\}$. Implementarea algoritmului de selecție quickselect (al k-lea element). Analiza comportamentului la worst-case al quicksort.</p>	2	Lucrare practică, benchmark	
<p>Lucrarea L13: Trie și aplicații text. Implementarea unui Trie pentru șiruri de caractere cu operații insert, search, delete, startsWith (prefix search). Aplicație: autocomplete — completarea unui prefix dat cu toate cuvintele stocate. Compararea timpului de căutare Trie vs. BST vs. hash set pentru un dicționar de 100K cuvinte.</p>	2	Lucrare practică individuală	
<p>Lucrarea L14: Proiect integrat și colocviu. Rezolvarea pe platforma online judge (LeetCode/Codeforces/PbInfo) a 3 probleme de dificultate medie care implică structuri de date și algoritmi din tematica disciplinei — câte una din: structuri de date (hash/heap/trie),</p>	2	Rezolvare online judge, colocviu oral	

grafuri (BFS/DFS/Dijkstra), DP sau divide-and-conquer. Prezentarea soluțiilor și justificarea alegerii algoritmului și a structurii de date. Colocviu oral pe baza soluțiilor prezentate.			
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Cormen, T.H. et al., Introduction to Algorithms (CLRS), 4th ed., MIT Press, 2022 — pseudocodul și demonstrațiile de corectitudine. 2. Sedgewick, R., Wayne, K., Algorithms, 4th ed. — implementările Java/Python și exercițiile: https://algs4.cs.princeton.edu/ 3. CP-Algorithms — implementări de referință în C++ pentru algoritmi din laborator: https://cp-algorithms.com/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Goodrich, M.T., Tamassia, R., Goldwasser, M.H., Data Structures and Algorithms in Python, Wiley, 2013 — implementările Python. 5. Skiena, S.S., The Algorithm Design Manual, 3rd ed., Springer, 2020 — problemele de proiectare. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. VisuAlgo — animații interactive pentru toate structurile și algoritmi din laborator: https://visualgo.net/ 7. LeetCode — platformă online judge cu probleme de structuri de date și algoritmi: https://leetcode.com/ 8. PbInfo — platformă românească de probleme de algoritmi: https://www.pbinfo.ro/ 9. Python Documentation — modulele heapq, collections, functools pentru structuri de date built-in: https://docs.python.org/3/library/ 		

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

<p>Structuri de Date și Algoritmi este una dintre disciplinele fundamentale ale programului TI, considerată de industria software drept piatra de temelie a competenței tehnice a oricărui inginer. Stăpânirea algoritmilor de sortare, a structurilor de date (heap, hash table, trie, union-find) și a algoritmilor pe grafuri (Dijkstra, Kruskal, DFS/BFS) este testată sistematic în interviurile tehnice ale companiilor din industria IT — de la startup-uri la giganti precum Google, Amazon sau Microsoft. Fără aceste competențe, studentul nu poate aborda eficient probleme din Machine Learning (KNN, k-means), baze de date (B-tree, hash join), rețele (routing, MST) sau compilatoare (grafuri de dependențe, tabele de simboluri).</p> <p>Conținuturile disciplinei sunt aliniate curriculum-ului ACM/IEEE CS2023 (Knowledge Area: Algorithms and Complexity, Data Structures and Algorithms) și acoperă integral materia testată în certificările tehnice pentru software engineering. Analiza riguroasă a complexității algoritmice — nu doar implementarea, ci și demonstrarea corectitudinii și justificarea alegerii algoritmului față de alternative — este competența centrală formată, reflectată direct în RÎ C8 și A6 din matricea de corelare.</p> <p>Lucrarea L14, bazată pe rezolvarea de probleme pe platforme online judge cu prezentare și justificare orală, reproduce fidel scenariul unui interviu tehnic real: studentul trebuie să aleagă structura de date și algoritmul adecvat, să justifice complexitatea soluției și să demonstreze că înțelege trade-off-urile față de soluții alternative. Această abordare formează nu doar competența de implementare, ci și capacitatea de a comunica transparent deciziile tehnice față de echipă și beneficiari — direct corelată cu R8 din matricea de corelare.</p>
--

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea principalelor structuri de date (liste, stive, cozi, arbori BST, heap, hash table, grafuri, trie) și a	Examen scris (2 ore): 1 problemă de analiză a complexității unui algoritm dat, 1 problemă de alegere și justificare a structurii de date optime pentru un scenariu	60%

	<p>operațiilor lor cu complexitățile corespunzătoare; capacitatea de a analiza complexitatea unui algoritm dat cu notații $O/\Omega/\Theta$; cunoașterea algoritmilor de sortare, a algoritmilor pe grafuri (DFS, BFS, Dijkstra, Kruskal) și a paradigmatelor (DP, divide-and-conquer, greedy).</p>	<p>descriși și 2 probleme de proiectare algoritmică (una din grafuri, una din DP sau divide-and-conquer) — cu pseudocod complet, demonstrare corectitudinii și analiza complexității</p>	
8.5 Seminar/laborator	<p>Corectitudinea și eficiența soluțiilor implementate la laborator (cod compilabil, teste trecute); calitatea rapoartelor (justificarea alegerii structurii/algoritmului, analiza complexității, benchmark-uri); calitatea soluțiilor și a prezentării la colocviul oral din L14; participarea activă la toate lucrările.</p>	<p>Evaluare continuă: soluții de laborator predate pe Sakai sau online judge după fiecare ședință, cu raport scurt de complexitate (50%) și colocviu oral la L14 — 3 probleme rezolvate pe online judge cu prezentare și justificare (50%)</p>	40%
8.6 Standard minim de performanță:			
<p>Rezolvarea corectă la examenul scris a cel puțin două din cele patru probleme, cu obligativitatea de a include în mod necesar problema de analiză a complexității — cu notația O corect justificată și nu doar enunțată. La colocviul oral din L14: acceptarea a cel puțin 2 din 3 soluții pe online judge cu dovada că testele sunt trecute (accepted) ȘI justificarea orală a complexității timp și spațiu a fiecărei soluții. Simpla funcționare a codului fără justificarea complexității nu îndeplinește standardul minim.</p>			

Data completării	Semnătura titularului de curs Lect. Dr. Lițan Daniela-Elena	Semnătura titularului de seminar / laborator Lect. Dr. Lițan Daniela-Elena
17.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		PROGRAMARE ORIENTATĂ PE OBIECTE – TIDS204					
1.2 Titularul activităților de curs		Lect. Dr. Panfiloiu Gheorghe					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Lect. Dr. Panfiloiu Gheorghe					
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Programarea Calculatoarelor și Limbaje de Programare I (TIDF105) — algoritmi, tipuri de date, funcții, tablouri; Programarea Calculatoarelor și Limbaje de Programare II (TIDF112) — structuri de date elementare, pointeri, alocare dinamică; Structuri de Date și Algoritmi (TIDS203) — liste, arbori, grafuri, complexitate.
3.2 de competențe	Stăpânirea programării procedurale în C/C++ sau Python (funcții, tablouri, pointeri/referințe, alocare dinamică de memorie, fișiere); cunoașterea noțiunilor de bază despre tipuri de date și structuri de control; familiarizarea cu un IDE (VS Code, CLion sau Eclipse) și cu compilatorul/interpretorul.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, diagramelor UML și temelor; recomandare: instalarea IDE-ului (IntelliJ IDEA Community pentru Java sau VS Code cu extensii C++/Python) și a
-------------------------------	--

	instrumentului de diagrame UML (draw.io sau PlantUML) înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; IDE instalat: IntelliJ IDEA Community Edition (Java) sau VS Code cu extensii C++17 sau Python 3.10+; JDK 17+ sau compilator GCC 11+; acces la GitHub pentru versionarea proiectelor de laborator; studenții pot folosi propriul laptop cu IDE-ul preferat.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C7. Deține cunoștințe specializate de programare structurată, orientată pe obiecte și funcțională, cu înțelegerea critică a paradigmatelor de programare și a domeniilor lor de aplicabilitate.
5.2 Aptitudini	A1. Aplică metode și tehnici avansate de proiectare orientată pe obiecte și utilizează șabloane de proiectare software (design patterns) pentru dezvoltarea de aplicații robuste, reutilizabile și ușor de întreținut. A3. Proiectează și implementează arhitecturi software aliniate cerințelor sistemului, asigurând integrarea și interoperabilitatea componentelor hardware și software într-un sistem informatic complex.
5.3 Responsabilitate și autonomie	R3. Își asumă responsabilitatea pentru calitatea codului sursă și a arhitecturii software proiectate, aplicând standarde și bune practici ingineresti și revizuire critic propriile soluții tehnice în raport cu cerințele de performanță și mentenabilitate.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în paradigma orientată pe obiecte. Limitele programării procedurale. Paradigme de programare: procedurală, OOP, funcțională, logică — comparație și domenii de aplicabilitate. Principiile fundamentale ale OOP: încapsulare, moștenire, polimorfism, abstractizare. Istoricul OOP: Simula, Smalltalk, C++, Java, Python. Avantaje OOP: reutilizabilitate, mentenabilitate, extensibilitate.	2	Prelegere interactivă, comparații	
Tema 2: Clase și obiecte. Clasa ca tip de date definit de utilizator. Câmpuri (atribute), metode (comportamente), constructori, destructori. Instanțierea obiectelor — alocarea în heap vs. stack. Modificatori de acces: public, private, protected. Metode getter/setter. Cuvântul cheie this. Membrii statici (câmpuri și metode de clasă). Implementare comparată în Java și C++.	2	Prelegere, exemple de cod, comparații	
Tema 3: Constructori și gestionarea resurselor. Constructori: implicit, parametrizat, de copiere. Lista de inițializare (C++). Destructorii în C++ vs. garbage collector în Java. Regula celor trei/cinci (C++): destructor, copy constructor, copy assignment, move constructor, move assignment. Semantica valorii vs. semantica referinței. Shallow copy vs. deep copy.	2	Prelegere, demonstrații cod	

<p>Tema 4: Moștenirea. Conceptul de moștenire — ierarhii de clase. Moștenire simplă și multiplă (C++). Moștenire în Java: extends. Suprascrierea metodelor (override). Cuvintele cheie super/base. Constructorii în ierarhii. Principiul substituției Liskov (LSP). Moștenire vs. compoziție — când se folosește fiecare. Clasa Object și metodele moștenite (toString, equals, hashCode în Java).</p>	2	Prelegere, diagrame UML, cod	
<p>Tema 5: Polimorfismul. Tipuri de polimorfism: ad-hoc (supraîncărcare), de subtyping (dinamic), parametric (generice). Legarea dinamică (dynamic dispatch) — tabela de metode virtuale (vtable) în C++. Cuvântul cheie virtual în C++ vs. @Override în Java. Polimorfismul prin pointeri/referințe la clasa de bază. Castarea: upcasting și downcasting. Operatorul dynamic_cast (C++) și instanceof (Java).</p>	2	Prelegere, analiză mecanisme	
<p>Tema 6: Clase abstracte și interfețe. Clase abstracte: metode abstracte (pure virtual în C++), instanțierea interzisă. Interfețe în Java: definiție, implementare multiplă, metode default și statice (Java 8+). Interfețe funcționale și expresii lambda (Java 8+). Diferența clasă abstractă vs. interfață. Design by contract: contractul definit prin interfețe. Principiul segregării interfețelor (ISP).</p>	2	Prelegere, exemple Java/C++	
<p>Tema 7: Excepții și tratarea erorilor. Mecanismul de excepții: throw, try-catch-finally (Java) / try-catch (C++). Ierarhia excepțiilor în Java: Throwable, Error, Exception, RuntimeException. Excepții checked vs. unchecked. Excepții personalizate. Principii de tratare a erorilor: fail-fast, nu ignora excepțiile, excepții specifice. Resurse cu try-with-resources (Java) și RAII (C++). Logging vs. propagare.</p>	2	Prelegere interactivă, bune practici	
<p>Tema 8: Generice și programare parametrică. Generice în Java: clase generice, metode generice, wildcards (? extends T, ? super T), type erasure. Template-uri în C++: class template, function template, template specialization. Avantajele genericității: type safety, eliminarea cast-urilor. Colecțiile generice: List, Map, Set în Java STL. Iterator pattern.</p>	2	Prelegere, exemple cod	
<p>Tema 9: Colecții și framework-uri de colecții. Java Collections Framework: interfețele List, Set, Queue, Map — implementări (ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap). Complexitatea operațiilor. Iteratori și for-each. C++ STL: vector, list, set, map, unordered_map. Alegerea colecției potrivite în funcție de cerințe. Algoritmi generici (sort, find, filter).</p>	2	Prelegere, benchmark-uri	

<p>Tema 10: Principiile SOLID și design patterns — introducere. Principiile SOLID: SRP, OCP, LSP, ISP, DIP — definiție, exemple de respectare și încălcare. Introducere în design patterns: categorii (creaționale, structurale, comportamentale). Pattern-uri creaționale: Singleton, Factory Method, Abstract Factory, Builder. Implementare și diagrame UML.</p>	2	Prelegere, diagrame UML	
<p>Tema 11: Design patterns — structurale și comportamentale. Pattern-uri structurale: Adapter, Decorator, Facade, Composite, Proxy. Pattern-uri comportamentale: Observer, Strategy, Template Method, Iterator, Command. Diagrame UML și implementări în Java/C++. Recunoașterea pattern-urilor în biblioteci standard (Iterator, Observer în JavaFX/Swing, Strategy în Comparator).</p>	2	Prelegere, diagrame UML, cod	
<p>Tema 12: Programare funcțională în context OOP. Lambda expressions în Java 8+ și C++11+. Interfețe funcționale: Predicate, Function, Consumer, Supplier. Stream API în Java: filter, map, reduce, collect. Funcții anonime și closures. Imutabilitate și efecte secundare. Combinarea OOP cu FP: avantaje și cazuri de utilizare. Optional în Java.</p>	2	Prelegere interactivă, exemple	
<p>Tema 13: Testarea unitară și calitatea codului OOP. Principii de cod curat (Clean Code — R. Martin): naming, funcții mici, comentarii. Code smells: God Object, Feature Envy, Long Method. Refactorizarea codului OOP. Testarea unitară cu JUnit 5 (Java) sau pytest (Python): adnotări, assertions, mock objects (Mockito). Test-Driven Development (TDD): ciclul red-green-refactor.</p>	2	Prelegere, demonstrații TDD	
<p>Tema 14: Modularizare, pachete și recapitulare. Organizarea codului în pachete/module (Java modules — JPMS, C++ namespaces). Vizibilitatea la nivel de pachet. Dependentele între module și principiul dependenței inverse (DIP). Injecția de dependențe (DI) — concept și utilitate. Recapitulare generală: de la clase și obiecte la arhitecturi software cu SOLID și design patterns.</p>	2	Prelegere, recapitulare, proiecte	

Bibliografie obligatorie

1. Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994 (retipărit 2022) — GoF, referința mondială pentru design patterns.
2. Bloch, J., Effective Java, 3rd ed., Addison-Wesley, 2018.
3. Martin, R.C., Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008 (retipărit 2020).

Bibliografie complementară

4. Martin, R.C., Agile Software Development: Principles, Patterns, and Practices, Prentice Hall, 2002 (retipărit 2020) — principiile SOLID.
5. Lippman, S.B., Lajoie, J., Moo, B.E., C++ Primer, 5th ed., Addison-Wesley, 2012 (retipărit 2021).
6. Horstmann, C.S., Core Java, Vol. 1 — Fundamentals, 12th ed., Oracle Press, 2021.

	Resurse software: 7. Refactoring Guru — catalogul complet al design patterns cu diagrame UML și cod Java/C++/Python: https://refactoring.guru/design-patterns 8. Oracle Java Documentation — Java SE 17 API și tutoriale oficiale: https://docs.oracle.com/en/java/javase/17/ 9. PlantUML — generare diagrame UML din text, gratuit: https://plantuml.com/		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Clase, obiecte și encapsulare. Implementarea unei clase complete (ex: BankAccount, Student, Rectangle) cu: câmpuri private, constructori supraîncărcați, getters/setters cu validare, metoda toString, metoda equals. Crearea de obiecte și testarea tuturor metodelor. Implementarea în Java (cu IntelliJ) sau C++ (cu VS Code).	2	Lucrare practică individuală	
Lucrarea L2: Moștenire și polimorfism dinamic. Proiectarea unei ierarhii de clase (ex: Shape → Circle, Rectangle, Triangle) cu metode abstracte (area(), perimeter(), toString()). Demonstrarea polimorfismului prin tablouri/liste de referințe la clasa de bază. Implementarea unei metode statice care operează polimorfic pe o colecție de forme.	2	Lucrare practică, diagrame UML	
Lucrarea L3: Interfețe, clase abstracte și design by contract. Definierea unei interfețe Comparable/Sortable și implementarea ei în mai multe clase. Implementarea Pattern-ului Strategy folosind interfețe pentru a selecta dinamic algoritmul de sortare sau de calcul. Testarea flexibilității față de extensibilitate (principiul OCP).	2	Lucrare practică individuală	
Lucrarea L4: Generice și colecții. Implementarea unei clase generice Stack sau Queue din primul principiu. Utilizarea Java Collections (ArrayList, HashMap, TreeMap) sau C++ STL (vector, map) pentru rezolvarea unor probleme concrete: frecvența cuvintelor dintr-un text, gruparea studenților după notă, cache LRU simplu.	2	Lucrare practică individuală	
Lucrarea L5: Design patterns — creaționale și structurale. Implementarea a două design patterns: Singleton (ex: Logger, ConfigManager) și Decorator (ex: adăugarea de funcționalități unui flux de I/O sau unui widget). Diagrama UML desenată cu PlantUML. Justificarea alegerii pattern-ului față de alternativa directă.	2	Lucrare practică, diagrame UML	
Lucrarea L6: Excepții, testare unitară cu JUnit/pytest și refactorizare. Adăugarea tratării excepțiilor personalizate la un proiect existent. Scrierea a 10 teste unitare (JUnit 5 sau pytest) pentru o clasă cu logică de business — acoperind cazuri normale, extreme și erori. Identificarea și corectarea a 2-3 code smells cu Refactoring Guru ca referință.	2	Lucrare practică, TDD	
Lucrarea L7: Proiect integrat OOP și colocviu. Implementarea unui mini-proiect OOP	2	Proiect individual, colocviu oral	

complet (ex: sistem de gestiune bibliotecă, magazin online simplificat, joc text-based) cu: minim 5 clase interconectate, cel puțin 2 design patterns aplicate, tratarea excepțiilor, persistență în fișier JSON/CSV, 5 teste unitare. Prezentarea diagramei UML și justificarea deciziilor arhitecturale. Colocviu oral.			
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> Gamma, E. et al., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994 — pseudocodul și diagramele UML. Bloch, J., Effective Java, 3rd ed., Addison-Wesley, 2018 — capitolele de clase, obiecte și generice. Refactoring Guru — implementări complete ale design patterns în Java, C++, Python: https://refactoring.guru/design-patterns <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> Martin, R.C., Clean Code, Prentice Hall, 2008 — capitolele de clase și design. Horstmann, C.S., Core Java, Vol. 1, 12th ed., Oracle Press, 2021 — exercițiile practice. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> Oracle Java Tutorials — colecții, generice, lambda, streams: https://docs.oracle.com/javase/tutorial/ JUnit 5 User Guide — documentație oficială pentru testare unitară Java: https://junit.org/junit5/docs/current/user-guide/ PlantUML Online — generare diagrame UML din text: https://www.plantuml.com/plantuml/uml/ GitHub Student Developer Pack — acces gratuit la IntelliJ IDEA Ultimate și alte unelte: https://education.github.com/pack 		

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

Programarea Orientată pe Obiecte este disciplina care face tranziția de la programarea procedurală la proiectarea de software real, de producție. Competențele formate — proiectarea claselor, aplicarea principiilor SOLID, utilizarea design patterns și scrierea de teste unitare — sunt direct solicitate în procesele de angajare din industria software, atât în interviurile tehnice, cât și în activitatea zilnică de development. Fără o înțelegere solidă a OOP, studentul nu poate aborda cursurile avansate din program: Ingineria Programelor, Proiectarea Algoritmilor, Aplicații Web sau Machine Learning, toate construite pe arhitecturi OOP.

Conținuturile disciplinei acoperă integral principiile din standardele industriale — SOLID (Robert C. Martin), Gang of Four design patterns și Clean Code — și sunt alinate curriculum-ului ACM/IEEE CS2023 (Software Engineering, Knowledge Area: Object-Oriented Design). Introducerea testării unitare (JUnit, TDD) și a refactorizării răspunde cerințelor angajatorilor care impun acoperirea codului cu teste ca practică obligatorie în echipele Agile.

Lucrarea L7 — proiectul integrat cu colocviu oral — reproduce un scenariu de evaluare a competenței tehnice din industrie: studentul prezintă arhitectura unui sistem OOP real, justifică deciziile de design (alegerea pattern-urilor, separarea responsabilităților, tratarea erorilor) și demonstrează că poate revizui critic propria soluție. Această abordare formează direct competența R3 din matricea de corelare: responsabilitatea pentru calitatea codului și a arhitecturii, aplicând standarde și bune practici ingineresti.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și aplicarea celor 4 principii OOP (encapsulare, moștenire, polimorfism,	Examen scris (2 ore): 1 subiect teoretic (identificarea principiilor SOLID respectate/încălcate într-un cod dat), 1 problemă de	60%

	abstractizare); înțelegerea principiilor SOLID și capacitatea de a identifica încălcările lor; cunoașterea a cel puțin 5 design patterns (GoF) cu diagrama UML și contextul de aplicare; capacitatea de a proiecta o ierarhie de clase pentru un scenariu dat.	proiectare OOP cu diagrama UML (minim 4 clase) și 1 problemă de design patterns (alegerea și implementarea unui pattern pentru un scenariu descris) — cu justificarea deciziilor	
8.5 Seminar/laborator	Corectitudinea și calitatea codului la laborator (compilabil, teste trecute, cod curat); calitatea proiectului integrat din L7 (arhitectură OOP coerentă, design patterns aplicate corect, teste unitare funcționale); calitatea prezentării și a justificărilor la colocviul oral din L7; participarea activă la toate lucrările.	Evaluare continuă: lucrări de laborator individuale predate pe GitHub/Sakai după fiecare ședință (50%) și proiect integrat cu colocviu oral individual la L7 — prezentarea arhitecturii, a design patterns aplicate și a testelor unitare (50%)	40%

8.6 Standard minim de performanță:

Implementarea în proiectul integrat din L7 a cel puțin două design patterns diferite (din categorii diferite: creațional + structural sau creațional + comportamental), cu diagrama UML corespunzătoare și justificarea orală a alegerii față de alternativa directă ȘI existența a minimum 5 teste unitare care trec (JUnit Accepted / pytest PASSED). Cod funcțional fără design patterns explicabile și fără teste nu îndeplinește standardul minim, indiferent de complexitatea implementării.

Data completării	Semnătura titularului de curs Lect. Dr. Panfiloiu Gheorghe	Semnătura titularului de seminar / laborator Lect. Dr. Panfiloiu Gheorghe
18.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	MATEMATICI DISCRETE – TIDS205						
1.2 Titularul activităților de curs	Lect. Dr. Antohe Valerian						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Lect. Dr. Ion Adelina						
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	3	din care: 2.2 curs	2	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	42	din care: 2.5 curs	28	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					33
Documentare					10
Studiu individual					10
Referate					10
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					3
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Analiză Matematică I (TIDF101), Algebră Liniară (TIDF102) — logică matematică elementară, raționament deductiv, mulțimi, relații; Programarea Calculatoarelor I (TIDF105) — algoritmi, tipuri de date, funcții recursive; Structuri de Date și Algoritmi (TIDS203) — grafuri, complexitate algoritmică.
3.2 de competențe	Capacitatea de a scrie și evalua demonstrații matematice elementare (inducție, reducere la absurd); cunoașterea noțiunilor de bază despre mulțimi, relații și funcții; abilitatea de a implementa algoritmi elementari; familiarizarea cu conceptul de graf și cu operații de bază pe grafuri.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă (pentru demonstrații), videoproiector și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, problemelor propuse și temelor; recomandare: instalarea Python (cu biblioteca networkx și matplotlib) sau SageMath pentru verificarea calculelor și vizualizarea grafurilor.
-------------------------------	---

4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; Python 3.10+ cu bibliotecile networkx, matplotlib, sympy instalate; studenții rezolvă probleme cu ajutorul calculatorului și verifică rezultatele analitice prin implementare; accesul la Wolfram Alpha sau SageMath este permis și recomandat.
--	---

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C2. Posedă cunoștințe specializate de matematici discrete, teoria probabilităților și statistică matematică, necesare proiectării algoritmilor și analizei datelor în sisteme informatice complexe. C10. Cunoaște limbajele formale, automatele și principiile translaatoarelor, ca fundament teoretic al proiectării compilatoarelor și al înțelegerii critice a limbajelor de programare moderne.
5.2 Aptitudini	A7. Aplică metode numerice și tehnici de modelare matematică pentru simularea și rezolvarea problemelor ingineresti, utilizând instrumente software specializate de calcul numeric.
5.3 Responsabilitate și autonomie	R2. Gestionează independent ciclul complet de dezvoltare al unui produs software — de la analiza cerințelor până la testare și livrare — demonstrând capacitatea de a anticipa riscurile tehnice și de a propune soluții alternative viabile.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Logică matematică și raționament formal. Propoziții și conectori logici: negație, conjuncție, disjuncție, implicație, echivalență. Tabele de adevăr. Tautologii, contradicții, contingente. Echivalențe logice fundamentale: legile De Morgan, distributivitate, absorbție. Forma normală conjunctivă (FNC) și disjunctivă (FND). Inferență logică: modus ponens, modus tollens, silogism. Aplicații în proiectarea circuitelor logice și a condițiilor din cod.	2	Prelegere interactivă, demonstrații	
Tema 2: Logică cu predicate și demonstrații matematice. Cuantificatori: universal (\forall) și existențial (\exists). Negarea predicatelor cu cuantificatori. Metode de demonstrație: directă, prin contrapozitie, prin contradicție (reducere la absurd), prin cazuri. Principiul inducției matematice: inducție simplă și tare. Inducție structurală. Demonstrarea formulelor pentru sume, produse și proprietăți ale algoritmilor.	2	Prelegere, demonstrații la tablă	
Tema 3: Teoria mulțimilor. Mulțimi și operații: reuniune, intersecție, diferență, diferență simetrică, complement, produs cartezian. Legile algebrei mulțimilor. Mulțimi finite și infinite: cardinalul, mulțimi numărabile și nenumărabile (diagonala lui Cantor). Mulțimea putere (puterea mulțimii). Principiul includerii și excluderii (PIE) și aplicații în combinatorică și analiza algoritmilor.	2	Prelegere interactivă, exemple	

<p>Tema 4: Relații și funcții. Relații binare: proprietăți (reflexivitate, simetrie, antisimetrie, tranzitivitate). Relații de echivalență: clase de echivalență, mulțimea factor. Relații de ordine: parțiale și totale, diagrame Hasse. Funcții: injective, surjective, bijective. Compoziția funcțiilor. Funcția inversă. Funcții importante în informatică: funcția floor, ceiling, logaritmul discret.</p>	2	Prelegere, diagrame, exemple	
<p>Tema 5: Combinatorică — numărare și metode. Principiul multiplicării și al adunării. Permutări, aranjamente, combinații — cu și fără repetiție. Coeficienți binomiali și teorema binomului. Identitățile lui Pascal. Coeficienți multinomiali. Principiul porumbeilor (pigeonhole principle). Aplicații în analiza algoritmilor randomizați, criptografie și teoria codurilor.</p>	2	Prelegere interactivă, exerciții	
<p>Tema 6: Recurențe și funcții generatoare. Relații de recurență: definiție, rezolvare prin metode directe (ecuația caracteristică). Recurențe liniare omogene cu coeficienți constanți: ordinul 1 și 2. Recurențe neomogene. Teorema Master pentru analiza algoritmilor divide-and-conquer. Funcții generatoare ordinare: calcul și aplicații la numărarea combinatorică. Numerele Fibonacci — analiză completă.</p>	2	Prelegere, rezolvare pas cu pas	
<p>Tema 7: Teoria grafurilor — fundamente. Definiții: graf, multigraf, pseudograf, graf orientat. Grade ale nodurilor: teorema handshaking. Subgrafuri și grafuri parțiale. Conexitate: componente conexe, grafuri biconexe, punți. Grafuri speciale: complete (K_n), bipartite ($K_{m,n}$), cicluri (C_n), arbori, hipercub. Izomorfismul grafurilor. Reprezentarea grafurilor: matrice de adiacență, liste de adiacență — complexitate comparată.</p>	2	Prelegere, vizualizări Python	
<p>Tema 8: Arbori. Definiție și proprietăți echivalente ale arborilor (fără cicluri, conex, $n-1$ muchii). Arbori cu rădăcină: parcurgeri (preordine, inordine, postordine, BFS). Arbori de acoperire minimă (spanning trees). Numărul arborilor etichetați pe n noduri — formula lui Cayley. Arbori binari: proprietăți de numărare, numere Catalan. Aplicații: expresii aritmetice, arbori de decizie, Huffman.</p>	2	Prelegere, demonstrații	
<p>Tema 9: Euler, Hamilton și colorarea grafurilor. Lanțuri și circuite Euler: teorema lui Euler, algoritmul Fleury, algoritmul lui Hierholzer. Lanțuri și cicluri Hamilton: condiții suficiente (teoremele Ore, Dirac). Problema comis-voiajorului (TSP) — complexitate NP-hard. Colorarea nodurilor: numărul cromatic $\chi(G)$, teorema celor 4 culori</p>	2	Prelegere, demonstrații, aplicații	

(enunț). Colorarea muchiiilor: teorema lui Vizing. Aplicații: planificarea examenelor, registre CPU.			
Tema 10: Grafuri planare și duale. Definiție și caracterizare: formula lui Euler ($V - E + F = 2$). Teorema lui Kuratowski: K_5 și $K_{3,3}$ ca subgrafuri minori. Algoritmul de testare a planarității. Grafuri duale. Aplicații: proiectarea circuitelor integrate, rețele fără intersecții de fire. Grafuri cu proprietăți speciale: grafuri de interval, grafuri de corzi, grafuri perfecte.	2	Prelegere interactivă	
Tema 11: Teoria numerelor și aritmetică modulară. Divizibilitate, cel mai mare divizor comun (algoritmul lui Euclid extins). Congruențe: proprietăți, aritmetică modulo n . Teorema chineză a resturilor (CRT). Micul teoremă al lui Fermat și teorema lui Euler ($\varphi(n)$). Inversul modular. Testul de primalitate Miller-Rabin (introducere). Aplicații directe în criptografie: RSA (schema principală), hashing.	2	Prelegere, demonstrații, aplicații cripto	
Tema 12: Algebră booleană și circuite logice. Algebre booleene: axiome, proprietăți, funcții booleene. Minimizarea funcțiilor booleene: metoda Karnaugh (2, 3, 4 variabile), metoda Quine-McCluskey. Circuite combinaționale: sumator, comparator, multiplexor. Circuite secvențiale: bistabile (SR, D, JK, T). Mașini de stare (FSM): Moore și Mealy — definiție și exemple. Legătura cu automatele finite din teoria limbajelor formale.	2	Prelegere, tabele Karnaugh	
Tema 13: Limbaje formale și automate. Gramatici formale: ierarhia Chomsky (tip 0-3). Expresii regulate: sintaxă, semantică, operații. Automate finite deterministe (DFA): definiție, funcția de tranziție, limbaj acceptat. Automate finite nedeterministe (NFA): teorema de echivalență cu DFA (construcția subset). Lema pumping pentru limbaje regulate. Minimizarea DFA. Legătura cu expresii regulate din programare (regex).	2	Prelegere, diagrame automate	
Tema 14: Automate stive și gramatici independente de context. Gramatici independente de context (CFG): derivări, arbori de derivare, ambiguitate. Limbaje independente de context: exemple ($L_n = a^n b^n$), lema pumping CFL. Automate pushdown (PDA): definiție, echivalența cu CFG. Forma Normală Chomsky (CNF). Algoritmul CYK de parsare. Aplicații: parsarea expresiilor matematice, compilatoare (analiza sintactică). Recapitulare generală.	2	Prelegere, recapitulare	

	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Rosen, K.H., Discrete Mathematics and Its Applications, 8th ed., McGraw-Hill, 2019. Sipser, M., Introduction to the Theory of Computation, 3rd ed., Cengage Learning, 2012 (retipărit 2022) — pentru limbaje formale și automate. Graham, R.L., Knuth, D.E., Patashnik, O., Concrete Mathematics: A Foundation for Computer Science, 2nd ed., Addison-Wesley, 1994 (retipărit 2021). <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Epp, S.S., Discrete Mathematics with Applications, 5th ed., Cengage Learning, 2019. Biggs, N.L., Discrete Mathematics, 2nd ed., Oxford University Press, 2002 (retipărit 2019). Diestel, R., Graph Theory, 5th ed., Springer, 2017. Disponibil gratuit la: https://diestel-graph-theory.com/ <p>Resurse software:</p> <ol style="list-style-type: none"> MIT OpenCourseWare 6.042J — Mathematics for Computer Science (note de curs, probleme și soluții complete): https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/ Wolfram MathWorld — Discrete Mathematics: https://mathworld.wolfram.com/topics/DiscreteMathematics.html NetworkX Documentation — biblioteca Python pentru grafuri: https://networkx.org/documentation/stable/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
<p>Lucrarea L1: Logică, mulțimi și inducție matematică cu Python. Implementarea în Python a calculatorului de tabele de adevăr (simpy.logic). Verificarea echivalențelor logice și identificarea tautologiilor. Implementarea demonstrației prin inducție a unor formule (sume, Fibonacci) cu verificare computațională pentru $n \in \{1..100\}$. Aplicarea principiului includerii-excluderii pentru calculul cardinalelor.</p>	2	Lucrare practică Python/sympy	
<p>Lucrarea L2: Combinatorică și recurențe. Implementarea funcțiilor pentru permutări, aranjamente, combinații (cu și fără repetiție) în Python. Calculul coeficienților binomiali cu triunghiul lui Pascal. Rezolvarea recurențelor liniare cu ecuația caracteristică — verificarea soluției prin calcul numeric. Implementarea și analiza empirică a complexității algoritmilor divide-and-conquer (Teorema Master).</p>	2	Lucrare practică Python	
<p>Lucrarea L3: Grafuri cu NetworkX — reprezentare și parcurgeri. Crearea și vizualizarea grafurilor orientate și neorientate cu networkx și matplotlib. Implementarea parcurgerilor BFS și DFS. Calculul componentelor conexe, al diametrului și al eccentricității. Testarea izomorfismului între grafuri mici. Generarea și analiza grafurilor speciale: complete, bipartite, aleatorii Erdős-Rényi.</p>	2	Lucrare practică Python/networkx	
<p>Lucrarea L4: Algoritmi pe grafuri — Euler, Hamilton, colorare. Implementarea algoritmului Hierholzer pentru circuitul Euler. Verificarea condiției Ore/Dirac pentru cicluri Hamilton pe grafuri mici. Colorarea greedy a nodurilor unui graf și calculul numărului cromatic aproximativ. Aplicație: planificarea examenelor fără conflict ca problemă de colorare a grafurilor.</p>	2	Lucrare practică Python/networkx	

Lucrarea L5: Aritmetică modulară și criptografie. Implementarea algoritmului lui Euclid extins pentru ccd și inversul modular. Implementarea Teoremei Chineze a Resturilor. Implementarea schemei RSA simplificată: generare chei (p, q mici), cifrare, descifrare, verificare. Testarea primalității cu metoda Miller-Rabin. Analiza complexității operațiilor modulare.	2	Lucrare practică Python	
Lucrarea L6: Algebră booleană și minimizarea Karnaugh. Implementarea minimizatorului de funcții booleene cu metoda Quine-McCluskey în Python. Verificarea echivalenței a două circuite logice prin tabele de adevăr. Simularea unui automat finit determinist (DFA) pentru un limbaj regulat dat. Construirea NFA și conversia sa în DFA echivalent (construcția subset — implementare pas cu pas).	2	Lucrare practică Python	
Lucrarea L7: Limbaje formale și automate — proiect integrat și colocviu. Implementarea unui parser simplu bazat pe CFG (gramatică independentă de context) pentru expresii aritmetice, cu construirea arborelui de derivare. Testarea gramaticii cu algoritmul CYK. Minimizarea unui DFA cu algoritmul Hopcroft. Colocviu oral: prezentarea soluțiilor din L1-L7 și justificarea matematică a algoritmilor implementați.	2	Proiect individual, colocviu oral	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Rosen, K.H., Discrete Mathematics and Its Applications, 8th ed., McGraw-Hill, 2019 — exercițiile și pseudocodul. 2. Sipser, M., Introduction to the Theory of Computation, 3rd ed., Cengage Learning, 2022 — capitolele de automate și limbaje formale. 3. NetworkX Documentation — tutoriale și referință API pentru grafuri în Python: https://networkx.org/documentation/stable/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. MIT OCW 6.042J — seturi de probleme rezolvate (Problem Sets cu soluții): https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/ 5. Diestel, R., Graph Theory, 5th ed., Springer, 2017 (gratuit online): https://diestel-graph-theory.com/ <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. SageMath — sistem de calcul algebric gratuit pentru matematici discrete: https://www.sagemath.org/ 7. Wolfram Alpha — verificarea calculelor de combinatorică, recurențe, aritmetică modulară: https://www.wolframalpha.com/ 8. Visualgo — vizualizarea grafurilor și a algoritmilor pe grafuri: https://visualgo.net/ 9. RegExr — testarea și vizualizarea expresiilor regulate: https://regexr.com/ 		

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

Matematicile Discrete furnizează fundamentul teoretic fără de care ingineria software nu poate depăși nivelul programării empirice. Logica matematică stă la baza proiectării circuitelor digitale și a raționamentului despre corectitudinea programelor; combinatorica și recurențele sunt instrumente indispensabile pentru analiza complexității algoritmilor; teoria grafurilor este direct aplicată în rețele de calculatoare, baze de date, compilatoare și inteligența artificială; aritmetica modulară și teoria numerelor sunt fundamentul criptografiei moderne; iar limbajele formale și automatele stau la baza compilatoarelor, analizei lexicale și a expresiilor regulate utilizate zilnic în programare.

Conținuturile disciplinei sunt alinate curriculum-ului ACM/IEEE CS2023 (Knowledge Area: Discrete Structures — DS) care identifică matematicile discrete drept una din cele mai importante fundamente teoretice ale ingineriei calculatoarelor. Disciplina pregătește în mod direct cursurile din semestrul următor și din anii următori: Limbaje Formale și Automate (care aprofundează automatele și gramaticile), Proiectarea Algoritmilor (care utilizează intensiv combinatorica și recurențele), Teoria Transmisiunii Informației (bazată pe algebră booleană și coduri) și Securitatea Informatică (bazată pe aritmetică modulară și criptografie).

Componenta de laborator, cu implementarea în Python a conceptelor matematice — de la verificarea tautologiilor și simularea automatelor la implementarea RSA și minimizarea DFA — urmărește consolidarea înțelegerii prin calculul direct și validarea computațională a rezultatelor analitice. Colocviul oral din L7 verifică nu memorarea mecanică a formulelor, ci capacitatea studentului de a lega raționamentul matematic de implementare și de a justifica alegerile algoritmice.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea conceptelor fundamentale din toate capitolele disciplinei: logică și demonstrații, combinatorică, recurențe, teoria grafurilor, aritmetică modulară, algebră booleană, limbaje formale și automate; capacitatea de a demonstra riguros propoziții cu inducție matematică; capacitatea de a proiecta și analiza DFA/NFA pentru limbaje regulate date.	Colocviu scris (1,5 ore): 4 probleme — câte una din: demonstrație prin inducție sau logică, combinatorică sau recurențe, teorie a grafurilor și aritmetică modulară sau automate — cu demonstrarea completă a calculelor și justificarea pașilor	60%
8.5 Seminar/laborator	Corectitudinea implementărilor Python din laborator (cod funcțional, rezultate corecte); calitatea justificărilor matematice din rapoartele de laborator; capacitatea demonstrată la colocviul oral din L7 de a explica legătura dintre conceptul matematic și implementarea sa; participarea activă la toate lucrările.	Evaluare continuă: lucrări de laborator individuale predate pe Sakai după fiecare ședință, cu cod și raport scurt de justificare matematică (50%) și colocviu oral individual la L7 — prezentarea soluțiilor și justificarea algoritmilor implementați (50%)	40%
8.6 Standard minim de performanță:			
Rezolvarea corectă și cu demonstrație completă la colocviul scris a cel puțin două din cele patru probleme, cu obligativitatea includerii problemei de demonstrație prin inducție matematică — aceasta verificând capacitatea de raționament formal, competența centrală a disciplinei. La colocviul oral din			

L7: implementarea funcțională a simulatorului DFA și explicarea orală a unui pas din construcția subset (NFA→DFA). Aplicarea formulelor fără demonstrarea pașilor de raționament nu îndeplinește standardul minim.

Data completării 18.03.2026	Semnătura titularului de curs Lect. Dr. Antohe Valerian	Semnătura titularului de seminar / laborator Lect. Dr. Ion Adelina
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei		PROIECTARE LOGICĂ – TIDS206					
1.2 Titularul activităților de curs		Prof. Dr. Mînză Viorel					
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator		Prof. Dr. Mînză Viorel					
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	Ob DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					94
Documentare					27
Studiu individual					26
Referate					26
Teme casă					5
Proiect					5
Alte activități (inclusive colocviu, examen)					5
2.9 Total ore pe semestru					150
2.10 Numărul de credite					6

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Matematici Discrete (TIDS205) — algebră booleană, porți logice, funcții booleene, minimizare Karnaugh; Electrotehnică (TIDS113) — circuite electrice, semnale digitale; Dispozitive Electronice (TIDS114) — tranzistoare ca comutatoare, niveluri logice; Fizică (TIDF103) — noțiuni de semiconductor.
3.2 de competențe	Cunoașterea și aplicarea algebrei booleene și a metodelor de minimizare (Karnaugh, Quine-McCluskey); cunoașterea funcționării porților logice fundamentale (AND, OR, NOT, NAND, NOR, XOR, XNOR) și a tabelelor lor de adevăr; abilitatea de a interpreta și desena scheme logice simple; familiarizarea cu utilizarea unui EDA (Electronic Design Automation) sau simulator de circuite.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoprojector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, fișilor de date ale componentelor și temelor; recomandare: instalarea
-------------------------------	---

	Logisim-Evolution (simulator gratuit de circuite logice) și a unui tool HDL (Quartus Prime Lite sau Vivado WebPACK) înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de electronică digitală cu calculatoare individuale; Logisim-Evolution instalat pentru simularea circuitelor combinaționale și secvențiale; Quartus Prime Lite sau Vivado WebPACK pentru sinteza HDL (FPGA); opțional: plăci de dezvoltare FPGA (DE10-Nano, Basys 3) sau kituri cu porți logice discrete (74xxx); acces la ModelSim pentru simularea HDL.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C5. Cunoaște în profunzime principiile proiectării logice și ale electronicii digitale, inclusiv arhitecturile pentru prelucrare numerică de semnal, situate la avangarda domeniului.
5.2 Aptitudini	A9. Proiectează și configurează sisteme de calcul bazate pe microprocesoare și procesoare digitale de semnal (DSP), integrând componente hardware și software pentru realizarea de sisteme embedded funcționale.
5.3 Responsabilitate și autonomie	R5. Își asumă responsabilitatea pentru coordonarea activităților tehnice ale unei echipe de dezvoltare software, distribuind sarcinile în mod echitabil, monitorizând progresul și intervenind proactiv pentru rezolvarea blocajelor apărute.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Sisteme de numerație și coduri. Sisteme de numerație poziționale: binar, octal, hexazecimal. Conversia între baze. Operații aritmetice în binar: adunare, scădere, înmulțire, împărțire. Reprezentarea numerelor întregi cu semn: complement față de 1, complement față de 2, semn-magnitudine. Depășirea (overflow). Coduri binare uzuale: BCD, Gray, ASCII, Unicode, coduri de detecție și corecție a erorilor (paritate, Hamming).	2	Prelegere interactivă, exerciții numerice	
Tema 2: Algebra booleană și funcții logice. Axiomele algebrei booleene (Huntington). Teoreme fundamentale: De Morgan, dualitate, absorbție, consensul. Funcții booleene: reprezentarea prin tabel de adevăr, forma canonică sumă de minterme (SOP) și produs de maxterme (POS). Notății: minterme, maxterme. Complementul unei funcții. Implementarea cu porți NAND și NOR universale.	2	Prelegere, demonstrații, scheme	
Tema 3: Minimizarea funcțiilor booleene. Metoda Karnaugh: 2, 3, 4 variabile — grupuri de 1 și 0, don't care. Obținerea SOP minim și POS minim. Metoda Quine-McCluskey: algoritm sistematic, implicați primi, acoperire minimă (tabelul de acoperire). Complexitatea metodei. Comparatie Karnaugh vs. Quine-McCluskey. Minimizarea cu condiții incomplet specificate.	2	Prelegere, tabele Karnaugh, exerciții	

<p>Tema 4: Circuite combinaționale — proiectare. Metodologia de proiectare a circuitelor combinaționale: specificație ->> tabel adevăr ->> minimizare ->> implementare ->> simulare. Circuite de bază: sumator semi (Half Adder), sumator complet (Full Adder), sumatoare pe n biți (ripple carry, carry lookahead). Comparatoare de magnitudine. Codificatoare și decodificatoare (BCD-7segmente, binare).</p>	2	Prelegere, proiectare circuite	
<p>Tema 5: Circuite combinaționale — multiplexoare și demultiplexoare. Multiplexor (MUX) 2:1, 4:1, 8:1: structură, tabele de adevăr, implementare. Utilizarea MUX ca generator universal de funcții booleene. Demultiplexor (DEMUX) și decodificatoare active-low. Circuite de selectare: barrel shifter, priority encoder. Implementarea funcțiilor logice cu MUX și decodificatoare.</p>	2	Prelegere, exemple proiectare	
<p>Tema 6: Hazarduri în circuite combinaționale și timp de propagare. Întârzierea de propagare (propagation delay) în porți logice: t_{pLH}, t_{pHL}, t_{pd}. Calculul timpului de propagare al unui circuit combinațional. Hazardul static-1, static-0 și dinamic: cauze și eliminare prin adăugarea termenilor redundanți. Consumul de putere: static (CMOS leakage) și dinamic. Tehnologii logice: TTL, CMOS, ECL — comparație.</p>	2	Prelegere, calcule timing	
<p>Tema 7: Bistabile — elemente de memorie. Bistabil SR latch: cu porți NOR și NAND, condiția interzisă. Bistabil D latch: funcționare transparentă. Bistabil D flipflop declanșat pe front: pozitiv și negativ. Bistabil JK master-slave: eliminarea hazardului de cursă. Bistabil T (Toggle). Parametri temporali: setup time, hold time, propagation delay, f_{clk_max}. Metastabilitate.</p>	2	Prelegere, diagrame timp	
<p>Tema 8: Circuite secvențiale — mașini de stare. Mașina de stare finită (FSM): definiție, diagrama de stare, tabelul de stare. Mașina Mealy vs. mașina Moore: diferențe, avantaje. Codificarea stărilor: codificare binară, one-hot, Gray. Metodologia de proiectare a circuitelor secvențiale sincrone. Analiza unui circuit secvențial dat: extragerea tabelului de stare și a diagramei.</p>	2	Prelegere, diagrame stare, proiectare	
<p>Tema 9: Proiectarea circuitelor secvențiale — exemple. Proiectarea unui numărător sincron modulo-n cu flipflop-uri D și JK: tabel de tranziție, ecuații de excitare. Numărătoare Johnson și inelari. Registre de deplasare (shift register): SISO, SIPO, PISO, PIPO. Registru cu feedback (LFSR) și aplicații în generarea numerelor pseudo-aleatoare și CRC.</p>	2	Prelegere, proiectare pas cu pas	

<p>Tema 10: Memorii digitale. Clasificarea memoriilor: ROM, PROM, EPROM, EEPROM, Flash. RAM static (SRAM): celula cu 6 tranzistoare, citire/scriere. RAM dinamic (DRAM): celula cu 1 tranzistor, refresh. Organizarea memoriei: cuvinte \times biți, decodificarea adresei. Ierarhia memoriei: registre, cache (L1/L2/L3), RAM, memorie secundară. FIFO și LIFO ca structuri de memorie hardware.</p>	2	Prelegere interactivă, scheme	
<p>Tema 11: Limbaje de descriere hardware (HDL) — VHDL/Verilog. Concepte HDL: descriere structurală, flux de date, comportamentală. Sintaxa Verilog de bază: module, wire, reg, assign, always. Modelarea circuitelor combinaționale în Verilog. Modelarea flipflop-urilor și a FSM în Verilog. Procesul de sinteză: de la HDL la netlist. Simularea cu testbench. Comparatie VHDL vs. Verilog.</p>	2	Prelegere, cod HDL, demonstrații	
<p>Tema 12: FPGA — arhitectură și programare. Arhitectura FPGA: CLB (Configurable Logic Block), Look-Up Table (LUT), flipflop, rutare, blocuri RAM, DSP. Fluxul de proiectare FPGA: HDL \rightarrow sinteză \rightarrow implementare \rightarrow bistream \rightarrow programare. Constrângeri: timing constraints, pin constraints. FPGA vs. ASIC vs. microcontroller: când se folosește fiecare. Platforme: Xilinx, Intel/Altera, Lattice.</p>	2	Prelegere interactivă, flow diagram	
<p>Tema 13: Arhitecturi pentru prelucrare numerică de semnal (DSP). Operații DSP: convoluție, filtrare FIR/IIR, FFT. Cerințe hardware: multiply-accumulate (MAC), pipeline, paralelism. Arhitectura unui DSP: unitate MAC dedicată, memorie duală (Harvard), unitate de adresare circulară. Implementarea unui filtru FIR pe FPGA: multiplicatoare paralele, sumatoare în cascadă. Procesoare DSP dedicate: TI C6000, SHARC.</p>	2	Prelegere, arhitecturi DSP	
<p>Tema 14: Arhitecturi avansate și recapitulare. Sisteme embedded cu FPGA: SoC (System-on-Chip), Zynq (ARM + FPGA). Comunicații digitale la nivel hardware: SPI, I2C, UART — implementare în FPGA. Procesoare soft: MicroBlaze, NIOS II. Tendințe actuale: FPGA în cloud (AWS F1), AI accelerators (Google TPU), RISC-V. Recapitulare generală — de la porți logice la sisteme complexe.</p>	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Mano, M.M., Ciletti, M.D., Digital Design with an Introduction to the Verilog HDL, VHDL, and SystemVerilog, 6th ed., Pearson, 2018. 2. Wakerly, J.F., Digital Design: Principles and Practices, 5th ed., Pearson, 2018. 3. Harris, D., Harris, S., Digital Design and Computer Architecture: RISC-V Edition, Morgan Kaufmann / Elsevier, 2022. <p>Bibliografie complementară</p>		

	<p>4. Palnitkar, S., Verilog HDL: A Guide to Digital Design and Synthesis, 2nd ed., Prentice Hall, 2003 (retipărit 2021).</p> <p>5. Chu, P.P., FPGA Prototyping by Verilog Examples: Xilinx Spartan-3 Version, Wiley-IEEE Press, 2008 (retipărit 2020).</p> <p>6. Mînză, V., Sisteme logice programabile, Ed. Tehnopress, 2005 — curs al titularului, disponibil la biblioteca universității.</p> <p>Resurse software:</p> <p>7. Logisim-Evolution — simulator gratuit de circuite logice combinaționale și secvențiale: https://github.com/logisim-evolution/logisim-evolution</p> <p>8. Intel Quartus Prime Lite — sinteză HDL gratuită pentru FPGA Intel/Altera: https://www.intel.com/content/www/us/en/software-kit/programmable/quartus-prime-lite.html</p> <p>9. HDLBits — exerciții online de Verilog cu evaluare automată: https://hdlbits.01xz.net/</p>		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Sisteme de numerație și coduri — exerciții practice. Conversia manuală și cu Python între binar, octal, hexazecimal și zecimal pentru numere întregi și fracționare. Implementarea în Python a reprezentării în complement față de 2 și detectarea overflow-ului. Generarea codului Gray și verificarea proprietăților. Calculul parității și al codului Hamming pentru un octet.	2	Lucrare practică, Python	
Lucrarea L2: Funcții booleene și minimizare Karnaugh în Logisim. Implementarea funcțiilor booleene în forma SOP și POS în Logisim-Evolution. Minimizarea manuală a funcțiilor cu 4 variabile cu harta Karnaugh (cu și fără don't care). Verificarea rezultatelor prin simulare în Logisim. Implementarea cu porți NAND universale a funcțiilor minime obținute.	2	Lucrare practică Logisim	
Lucrarea L3: Circuite combinaționale — sumator și comparator. Proiectarea și simularea în Logisim a unui sumator complet pe 4 biți (ripple carry adder). Proiectarea unui comparator de magnitudine pe 4 biți. Măsurarea timpului de propagare în Logisim (simulare timp). Compararea ripple carry adder cu carry lookahead adder la nivel de delay.	2	Lucrare practică Logisim	
Lucrarea L4: Multiplexoare, decodificatoare și implementarea funcțiilor. Implementarea unui MUX 4:1 din porți logice și verificarea prin simulare. Utilizarea MUX 8:1 pentru implementarea unei funcții booleene arbitrare de 3 variabile. Proiectarea unui decodificator BCD-7segmente. Implementarea unui codificator cu prioritate (priority encoder) pe 4 intrări.	2	Lucrare practică Logisim	
Lucrarea L5: Bistabile și parametri temporali. Proiectarea în Logisim a unui SR latch, D latch și D flipflop declanșat pe front. Observarea condiției interzise la SR latch. Măsurarea setup time și hold time prin simulare temporală. Proiectarea unui registru de 8 biți cu D flipflop-uri și semnale de reset sincron/asincron.	2	Lucrare practică Logisim, timing	

<p>Lucrarea L6: Mașini de stare — proiectare și simulare. Proiectarea completă a unui FSM Moore pentru un detector de secvență (ex: detectarea șirului '1011' în flux serial): diagrama stare ->> tabel tranziții ->> ecuații excitare ->> circuit ->> simulare Logisim. Proiectarea unui FSM Mealy echivalent și compararea numărului de stări.</p>	2	Lucrare practică Logisim, FSM	
<p>Lucrarea L7: Numărătoare și registre de deplasare. Proiectarea unui numărător sincron modulo-10 (BCD) cu flipflop-uri D în Logisim. Implementarea unui numărător up/down cu load sincron. Proiectarea unui registru de deplasare pe 8 biți (PIPO, SIPO). Implementarea unui LFSR de 8 biți și analiza secvenței generate (perioadă, proprietăți pseudo-aleatoare).</p>	2	Lucrare practică Logisim	
<p>Lucrarea L8: Introducere în Verilog — circuite combinaționale. Primul program Verilog: un modul pentru un sumator complet cu testbench. Modelarea circuitelor combinaționale: assign (flux de date) și always @(*) (comportamental). Simularea cu Quartus Prime Lite / ModelSim. Scrierea unui testbench complet cu verificarea automată a rezultatelor (self-checking testbench).</p>	2	Lucrare practică Verilog/Quartus	
<p>Lucrarea L9: Verilog — circuite secvențiale și FSM. Implementarea în Verilog a unui D flipflop cu reset asincron și sincron. Implementarea unui numărător modulo-N parametrizabil. Implementarea în Verilog a unui FSM Moore pentru detectarea unei secvențe — compararea cu implementarea din Logisim (L6). Simulare și verificare cu testbench.</p>	2	Lucrare practică Verilog/Quartus	
<p>Lucrarea L10: Verilog — memorii și comunicații seriale. Implementarea în Verilog a unui RAM sincron simplu (16×8 biți) cu citire și scriere. Implementarea unui UART TX simplu (transmițător serial) în Verilog: conversia paralel-serie, bit de start, stop, baud rate. Testarea cu ModelSim și vizualizarea semnalelor în formă de undă.</p>	2	Lucrare practică Verilog/ModelSim	
<p>Lucrarea L11: Sinteză pe FPGA — proiect combinațional. Implementarea completă pe FPGA (sau în Quartus/Vivado cu simulare) a unui calculator aritmetic pe 4 biți: adunare, scădere, AND, OR cu selector de operație pe 2 biți și afișare pe LED-uri sau display 7-segmente. Atribuirea pinurilor, constrângeri de timing, programarea plăcii (dacă disponibilă).</p>	2	Lucrare practică FPGA/Quartus	
<p>Lucrarea L12: Sinteză pe FPGA — proiect secvențial. Implementarea unui sistem de semaforizare cu FSM Moore pe FPGA: 3 stări (roșu, galben, verde) cu timere implementate</p>	2	Lucrare practică FPGA/Quartus	

cu numărătoare. Afișarea stării pe LED-uri RGB. Adăugarea unui buton de forțare a stării de urgență. Verificarea constrângerilor de timing post-sinteză.			
Lucrarea L13: HDLBits și exerciții Verilog avansate. Rezolvarea a 5 probleme de dificultate medie de pe platforma HDLBits (https://hdlbits.01xz.net/): circuite combinaționale, FSM, operații pe vectoare, generarea semnalelor de ceas. Implementarea unui filtru median pe 3 valori în Verilog cu logică combinațională. Compararea diferitelor stiluri de descriere HDL.	2	Lucrare practică online, Verilog	
Lucrarea L14: Proiect integrat și colocviu. Proiectarea și implementarea completă în Verilog a unui sistem digital cu: circuit combinațional + circuit secvențial (FSM) + memorie simplă, la alegerea studentului (ex: joc Simon, controller SPI/I2C, calculator simplu cu afișaj). Prezentarea diagramelor (schema bloc, diagrama de stare), a codului HDL și a simulărilor. Colocviu oral pe baza proiectului.	2	Proiect individual, colocviu oral	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> 1. Mano, M.M., Ciletti, M.D., Digital Design with an Introduction to the Verilog HDL, 6th ed., Pearson, 2018 — exercițiile de proiectare. 2. HDLBits — exerciții Verilog cu evaluare automată: https://hdlbits.01xz.net/ 3. Logisim-Evolution — documentație și tutoriale: https://github.com/logisim-evolution/logisim-evolution <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> 4. Chu, P.P., FPGA Prototyping by Verilog Examples, Wiley-IEEE Press, 2008 — implementările practice pe FPGA. 5. Harris, D., Harris, S., Digital Design and Computer Architecture: RISC-V Edition, Morgan Kaufmann, 2022 — laboratoarele Verilog. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. Logisim-Evolution — simulator circuite logice combinaționale și secvențiale: https://github.com/logisim-evolution/logisim-evolution 7. Intel Quartus Prime Lite Edition — sinteză HDL gratuită: https://www.intel.com/content/www/us/en/software-kit/programmable/quartus-prime-lite.html 8. HDLBits — exerciții Verilog cu simulare online: https://hdlbits.01xz.net/ 9. EDA Playground — simulare HDL online (Verilog/VHDL): https://www.edaplayground.com/ 		

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

Proiectarea Logică furnizează fundamentul hardware al întregului program de studii TI: fără înțelegerea modului în care funcționează porțile logice, flipflop-urile, mașinile de stare și memoriile, studentul nu poate înțelege cu adevărat cum funcționează un procesor, un microcontroller sau un SoC. Disciplina creează legătura esențială dintre nivelul fizic al electronică digitale și nivelul architectural al calculatoarelor, pregătind direct cursurile de Arhitectura Sistemelor de Calcul, Proiectarea cu Microprocesoare și Sisteme Embedded.

Conținuturile acoperă curriculum-ul IEEE/ACM CE2016 pentru Computer Engineering (Knowledge Area: Digital Design), inclusiv limbajele de descriere hardware (Verilog) și tehnologiile FPGA, care sunt standardul industrial pentru prototiparea sistemelor digitale. Introducerea arhitecturilor DSP (Tema 13) răspunde cerinței C5 din matricea de corelare — cunoașterea arhitecturilor pentru prelucrare numerică de semnal — și creează un pod natural cu disciplinele de semnal și comunicații din semestrul următor.

Proiectul integrat din L14 — un sistem digital complet proiectat în Verilog, cu circuit combinațional, FSM și memorie — reproducea la scară mică un proiect real de inginerie hardware: de la specificație la schemă bloc, cod HDL, simulare și demonstrație funcțională. Prezentarea și justificarea deciziilor de proiectare în fața titularului formează nu doar competența tehnică, ci și capacitatea de a coordona și explica deciziile tehnice ale echipei (R5 din matricea de corelare).

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea sistemelor de numerație și a reprezentărilor numerice; stăpânirea algebrei booleene și a metodelor de minimizare (Karnaugh, Quine-McCluskey); capacitatea de a proiecta circuite combinaționale (sumator, comparator, MUX, decodificator) și secvențiale (FSM Moore/Mealy, numărătoare, registre); cunoașterea sintaxei Verilog și a arhitecturii FPGA.	Examen scris (2 ore): 1 problemă de minimizare (Karnaugh cu don't care), 1 problemă de proiectare circuit combinațional (cu derivarea ecuațiilor și schema logică), 1 problemă de proiectare FSM (diagrama de stare, tabel de tranziție, ecuații de excitare) și 1 problemă de Verilog (scrierea unui modul cu testbench) — cu bareme detaliate	60%
8.5 Seminar/laborator	Corectitudinea simulărilor în Logisim și a implementărilor Verilog din laborator; calitatea proiectului integrat din L14 (funcționalitate verificată prin simulare, cod HDL curat, documentație schema bloc + diagrama stare); calitatea prezentării și a justificărilor tehnice la colocviul oral din L14; participarea activă la toate lucrările.	Evaluare continuă: lucrări de laborator individuale predate pe Sakai după fiecare ședință, cu fișierele de simulare și un raport scurt de proiectare (50%) și proiect integrat cu colocviu oral individual la L14 — sistem digital complet în Verilog cu prezentare de 10 minute (50%)	40%
8.6 Standard minim de performanță:			
Proiectarea corectă și completă la examenul scris a circuitului secvențial (FSM) — cu diagrama de stare, tabelul de tranziție și ecuațiile de excitare derivate riguros, fără erori de sinteză — aceasta verificând competența centrală a disciplinei ȘI demonstrarea funcționalității proiectului integrat din L14 prin simulare HDL cu testbench (cel puțin 3 cazuri de test documentate). Proiect prezentat fără simulare verificabilă sau fără justificarea deciziilor de proiectare nu îndeplinește standardul minim.			

Data completării 18.03.2026	Semnătura titularului de curs Prof. Dr. Mînză Viorel	Semnătura titularului de seminar / laborator Prof. Dr. Mînză Viorel
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	LIMBA STRĂINĂ III – TIDC207						
1.2 Titularul activităților de curs	Conf. Dr. Cheșcă Alina-Beatrice						
1.3 Titularul activităților de seminar	Conf. Dr. Cheșcă Alina-Beatrice						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Ob DC

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

2.1 Număr de ore pe săptămână	2	din care: 2.2 curs	1	2.3 seminar/laborator	1/0
2.4 Total ore din planul de învățământ	28	din care: 2.5 curs	14	2.6 seminar/laborator	14/0
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					22
Documentare					5
Studiu individual					5
Referate					5
Teme casă					3
Proiect					2
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					50
2.10 Numărul de credite					2

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Limba Străină I (TIDF107) și Limba Străină II (TIDC115) — nivel B2 consolidat: vocabular tehnic IT, comunicare profesională scrisă și orală, prezentări tehnice structurate, documentație tehnică.
3.2 de competențe	Nivel B2 CEFR consolidat în limba engleză: capacitatea de a susține o prezentare tehnică structurată de cel puțin 5 minute, de a redacta documente tehnice la nivel de paragraf și de a participa activ la discuții profesionale pe teme IT; vocabularul tehnic acumulat în semestrul I și II.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector, tablă albă și acces la internet; platforma e-learning Danubius Online (Sakai) pentru distribuirea materialelor de curs, textelor autentice și temelor; acces la resurse Cambridge/Oxford și la articole tehnice IEEE Spectrum, ACM Communications.
4.2 de desfășurare a seminarului/laboratorului	Sală de seminar cu acces la internet și prezentări multimedia; studenții sunt încurajați să utilizeze dispozitivele proprii pentru accesul la resurse online; activitățile de seminar includ simulări de prezentări tehnice, jocuri de rol (interviu tehnic, ședință de proiect), peer feedback structurat și lucru cu texte autentice de specialitate.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	Deține cunoștințe consolidate de limbă engleză la nivel B2+/C1 (CEFR), cu vocabular tehnic specializat extins pentru domenii de avangardă: arhitecturi software avansate, DevOps și livrare continuă, dezvoltare mobilă nativă și hibridă, securitate informatică și comunicare tehnică în contexte academice și profesionale internaționale.
5.2 Aptitudini	Elaborează și susține prezentări tehnice complexe pentru audiențe cu profiluri mixte — tehnic și non-tehnic — adaptând conținutul, nivelul de detaliu și suportul vizual la specificul publicului; redactează cu acuratețe genuri de text tehnic avansat (rapoarte, whitepapers, articole academice, post-mortems) și comunică fluent în toate situațiile profesionale specifice inginerului TI: interviuri tehnice, negocieri, ședințe de proiect, conferințe internaționale.
5.3 Responsabilitate și autonomie	Utilizează limba engleză cu deplină responsabilitate și autonomie în comunicarea profesională tehnică, asumându-și răspunderea pentru acuratețea terminologiei în documentele tehnice și pentru calitatea mesajului transmis; promovează o cultură a transparenței și a preciziei în comunicarea cu echipe multiculturale distribuite, respectând normele de integritate academică și profesională în producerea și citarea conținutului tehnic în engleză.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Technical Presentations — structură și livrare. Structura unei prezentări tehnice profesionale: opening hook, agendă, body, Q&A, closing. Limbajul de semnalizare (signposting): introducerea temelor, tranziții, rezumat, încheiere. Tehnici de captare și menținere a atenției audienței mixte. Analiza prezentărilor TED din domeniul IT: structură, vocabular, ritm.	2	Prelegere interactivă, analiză video	
Tema 2: Advanced Technical Writing — rapoarte și whitepapers. Structura unui raport tehnic: executive summary, methodology, findings, recommendations. Stilul tehnic avansat: precizie, concizie, voce pasivă vs. activă în context tehnic, nominalizare. Whitepaper-ul IT: definiție, audiență, structură. Analiza unui whitepaper real (AWS, Google Cloud, NIST).	2	Prelegere interactivă, analiză texte	
Tema 3: Describing complex systems and architectures. Vocabular pentru descrierea arhitecturilor software avansate: microservices, distributed systems, DevOps pipeline, CI/CD, containerization. Tehnici de descriere: comparație, cauză-efect, descrierea proceselor. Utilizarea diagramelor și a limbajului vizual în prezentări tehnice complexe.	2	Prelegere interactivă, exemple autentice	
Tema 4: Professional communication in international IT teams. Comunicarea asincronă: documentație tehnică clară (README, Wiki, runbook), comentarii de cod, pull request descriptions. Comunicarea în	2	Prelegere interactivă, studii de caz	

echipe multiculturale: directness vs. indirectness, culturi high-context vs. low-context. Gestionarea dezacordului tehnic constructiv. Comunicarea în criză: incident reports, post-mortems.			
Tema 5: Presenting data and research findings. Vizualizarea datelor: descrierea graficelor și diagramelor cu vocabular specific (spike, plateau, fluctuate, correlate). Prezentarea metricilor tehnice (latency, throughput, accuracy, precision/recall). Limbajul hedging pentru limitele cercetării. Structura unui abstract academic tehnic.	2	Prelegere interactivă, exerciții de descriere	
Tema 6: Job interviews and professional networking in English. Pregătirea pentru interviul tehnic în engleză: behavioral questions (metoda STAR), technical questions, negocierea salariului. CV-ul tehnic și profilul LinkedIn în engleză. Networking la conferințe tehnice: elevator pitch, follow-up e-mail. Mock interview: simulare completă cu feedback.	2	Prelegere, joc de rol, peer feedback	
Tema 7: Academic and professional writing — nivel avansat. Structura unui articol tehnic: abstract, introduction, related work, methodology, results, conclusion. Citarea surselor în engleză (IEEE style, APA). Parafrazarea și sinteza surselor multiple. Evitarea plagiatului. Procesul de peer review și revizie.	1	Prelegere interactivă, exerciții de scriere	
Tema 8: Sinteză și pregătire pentru colocviu. Recapitulare vocabular tehnic avansat acumulat în semestrul 3. Strategii pentru colocviu: gestionarea întrebărilor dificile în prezentarea orală, structurarea răspunsurilor la întrebările scrise. Autoevaluare cu grila CEFR B2+/C1. Feedback final și recomandări individuale.	1	Recapitulare, autoevaluare, feedback	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Ibbotson, M., Cambridge English for Engineering, Cambridge University Press, 2008. 2. Glendinning, E.H., McEwan, J., Basic English for Computing, Oxford University Press, 2002 (retipărit 2010). <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 3. Swales, J.M., Feak, C.B., Academic Writing for Graduate Students, 3rd ed., University of Michigan Press, 2012. 4. Powell, M., Dynamic Presentations, Cambridge University Press, 2010. 5. Mascull, B., Business Vocabulary in Use: Advanced, 3rd ed., Cambridge University Press, 2017. <p>Resurse software:</p> <ol style="list-style-type: none"> 6. IEEE Spectrum — articole tehnice autentice pentru lectură și vocabular de specialitate: https://spectrum.ieee.org/ 7. ACM Communications — studii de caz și articole IT la nivel avansat: https://cacm.acm.org/ 8. Cambridge Dictionary Online — dicționar englez cu exemple de uz tehnic și academic: https://dictionary.cambridge.org/ 		
6.2 Seminar	Nr. ore	Metode de predare	Observații
Tema S1: Presentation skills — rondul I de practică. Fiecare student susține o mini-prezentare tehnică de 3 minute pe o temă IT	2	Prezentări individuale, peer feedback	

din actualitate. Feedback structurat pe grilă: structură, claritate, vocabular, contact vizual, gestionarea întrebărilor. Identificarea punctelor forte și a ariilor de îmbunătățire pentru fiecare student.			
Tema S2: Technical report writing workshop. Atelier de scriere: redactarea unui executive summary de 150 de cuvinte pentru un proiect tehnic fictiv. Corectarea erorilor tipice în scrierea tehnică formală (false friends, register mixing). Analiza unui raport tehnic real și identificarea structurii și a marcatorelor discursivi.	2	Atelier de scriere, peer review	
Tema S3: Describing architectures and systems. Joc de rol: un inginer explică arhitectura unui sistem cloud unui client non-tehnic. Exerciții de simplificare a limbajului fără pierderea preciziei tehnice. Construirea unui glosar bilingval de termeni arhitecturali. Tehnica de descriere pe niveluri: overview — component — detaliu.	2	Joc de rol, exerciții practice	
Tema S4: Data presentation and infographics. Audierea și analiza unui segment dintr-o conferință tehnică (Google I/O, AWS re:Invent). Exerciții de descriere orală a graficelor și diagramelor cu vocabular specific. Redactarea unui paragraf de Results pentru un set de date dat (latență, throughput, acuratețe).	2	Ascultare activă, exerciții de scriere	
Tema S5: Mock technical interview. Simularea completă a unui interviu tehnic în engleză: behavioral questions (metoda STAR), technical explanation question, culture fit questions. Feedback individualizat pe vocabular, fluentă și structura răspunsului.	2	Simulare interviu, feedback individual	
Tema S6: Academic writing — sinteză și parafrază. Exerciții de parafrază a unor paragrafe din articole IEEE/ACM. Sinteza a două surse contradictorii pe o temă tehnică. Redactarea unui related work section de 200 de cuvinte. Verificarea coeziunii și a coerenței textului academic.	2	Atelier de scriere, analiză text	
Tema S7: Cross-cultural communication simulation. Studiu de caz: un proiect software cu echipe distribuite (România, India, Germania, SUA). Analiza unui e-mail ambiguu și reformularea lui. Exercițiu de scriere a unui post-mortem report — echilibrul dintre transparență și diplomatie profesională.	1	Studiu de caz, scriere colaborativă	
Tema S8: Prezentări finale și evaluare portofoliu. Fiecare student susține prezentarea finală de portofoliu: 5 minute pe o temă tehnică aleasă, cu Q&A de 2 minute. Evaluare pe grilă cu criterii explicite. Feedback colectiv și individual. Autoevaluare finală cu grila CEFR.	1	Prezentări individuale, evaluare finală	

	<p>Bibliografie obligatorie seminar</p> <ol style="list-style-type: none"> 1. Ibbotson, M., Cambridge English for Engineering, Cambridge University Press, 2008 — activitățile de speaking și writing. 2. Powell, M., Dynamic Presentations, Cambridge University Press, 2010 — tehnicile de prezentare. 3. Swales, J.M., Feak, C.B., Academic Writing for Graduate Students, 3rd ed., University of Michigan Press, 2012 — exercițiile de scriere academică. <p>Bibliografie complementara seminar</p> <ol style="list-style-type: none"> 4. Murphy, R., English Grammar in Use, 5th ed., Cambridge University Press, 2019. 5. Mascull, B., Business Vocabulary in Use: Advanced, 3rd ed., Cambridge University Press, 2017. <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> 6. TED Talks (Technology) — prezentări autentice C1 cu subtitrare în engleză: https://www.ted.com/topics/technology 7. IEEE Spectrum — articole tehnice autentice pentru lectură și analiză: https://spectrum.ieee.org/ 8. Grammarly Blog — ghiduri de scriere profesională și academică în engleză: https://www.grammarly.com/blog/ 9. Coursera — Technical Writing certificate (Google): https://www.coursera.org/learn/technical-writing
--	---

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

<p>Limba Străină III marchează tranziția de la competența comunicativă B2 consolidată la comunicarea tehnică avansată la nivel B2+/C1 — nivelul solicitat în activitatea profesională reală a inginerului TI din industria internațională. Capacitatea de a prezenta un produs, o arhitectură sau rezultatele unui proiect în fața unor audiențe mixte, de a redacta documentație tehnică de calitate academică și de a comunica eficient în echipe multiculturale distribuite reprezintă competențe transversale recunoscute de angajatori ca factori decisivi de promovare în carieră.</p> <p>Conținuturile disciplinei sunt aliniate descriptorilor CEFR pentru nivelul B2+/C1 și acoperă genurile de text solicitate în activitatea zilnică a inginerului TI: rapoarte tehnice, whitepapers, articole academice, post-mortems, prezentări la conferințe și interviuri tehnice. Componenta de scriere academică (Temele 7 și S6) pregătește studentul atât pentru redactarea lucrării de licență în engleză, cât și pentru publicarea ulterioară în reviste internaționale cu recenzori.</p> <p>Structura progresivă a seminarului — cu două runde de prezentări orale (S1 și S8) și mai multe ateliere de scriere tehnică (S2, S6) — urmărește construirea treptată a încrederii și a competenței, cu feedback explicit după fiecare activitate. Prezentarea finală de portofoliu din S8, evaluată pe grilă CEFR transparentă, certifică progresul individual al studentului pe parcursul celor trei semestre de studiu al limbii engleze în program.</p>

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	<p>Cunoașterea structurii și a convențiilor genurilor de text tehnic avansat (raport tehnic, whitepaper, articol academic, post-mortem); însușirea vocabularului tehnic IT la nivel B2+/C1 (arhitecturi avansate, DevOps, mobile, cloud, securitate); cunoașterea tehnicilor de prezentare orală pentru audiențe mixte și a strategiilor de</p>	<p>Colocviu scris (1 oră): redactarea unui executive summary de 150 de cuvinte pentru un proiect tehnic dat și răspunsul la 10 întrebări de vocabular tehnic și structuri lingvistice din tematica cursului</p>	40%

	adaptare a mesajului la audiență.		
8.5 Seminar/laborator	Calitatea participării active la seminar; progresul demonstrat în prezentările orale (S1 vs. S8); calitatea temelor de scriere tehnică predate; calitatea prezentării finale de portofoliu din S8 și capacitatea de a gestiona Q&A în engleză.	Evaluare continuă: participare activă și teme scrise de seminar (50%) și prezentare finală de portofoliu la S8 — 5 minute pe o temă tehnică aleasă, cu Q&A de 2 minute, evaluată pe grilă CEFR cu criterii explicite (50%)	60%

8.6 Standard minim de performanță:

Suținerea prezentării finale de portofoliu la S8 cu un discurs coerent și fluent de minimum 5 minute în engleză pe o temă tehnică aleasă, demonstrând capacitatea de a adapta simultan nivelul de detaliu tehnic la o audiență mixtă (tehnic și non-tehnic) ȘI obținerea unui punctaj de minimum 50% la colocviul scris. Spre deosebire de semestrul II, prezentarea din S8 impune obligatoriu o componentă de vizualizare a datelor sau a arhitecturii descrise. Predarea a cel puțin 5 din 8 teme de seminar scrise constituie condiție obligatorie pentru accesul la colocviu.

Data completării	Semnătura titularului de curs Conf. Dr. Cheșcă Alina-Beatrice	Semnătura titularului de seminar / laborator Conf. Dr. Cheșcă Alina-Beatrice
17.03.2026		
Data avizării în Consiliul Departamentului	Semnătura directorului DIT Lect. dr. Antohe Valerian	
24.03.2026		
Data aprobării în Consiliul Facultății	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	
07.04.2026		



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	EDUCAȚIE FIZICĂ ȘI SPORT III – TIDC208						
1.2 Titularul activităților de curs							
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Prof. Dr. Ploesteanu Constantin						
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	A/R	1.8 Regimul disciplinei	Ob DC

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

2.1 Număr de ore pe săptămână	1	din care: 2.2 curs	-	2.3 seminar/laborator	0/1
2.4 Total ore din planul de învățământ	14	din care: 2.5 curs	-	2.6 seminar/laborator	0/14
Distribuția fondului de timp					ore
2.7 Activități monitorizate					
Practică					
Elaborare proiect diplomă					
2.8 Activități individuale					11
Documentare					3
Studiu individual					3
Referate					
Teme casă					
Proiect					
Alte activități (inclusive colocviu, examen)					5
2.9 Total ore pe semestru					25
2.10 Numărul de credite					1

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Educație Fizică și Sport I (TIDF108) și Educație Fizică și Sport II (TIDC116) — deprinderi motrice de bază și aplicativ-sportive consolidate; cunoașterea regulamentelor sporturilor practicate; condiție fizică generală evaluată și îmbunătățită pe parcursul anului I.
3.2 de competențe	Participare regulată la activitățile din semestrul I și II; nivel de condiție fizică corespunzător vârstei și sexului; absența unor contraindicații medicale pentru efort fizic de intensitate medie; dorința de a practica activitate fizică organizată în mod voluntar și responsabil.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Nu este cazul — disciplina nu are activitate de curs.
4.2 de desfășurare a seminarului/laboratorului	Sală de sport sau teren de sport al Universității Internaționale Danubius; echipament sportiv adecvat (tricou, pantalon scurt, încălțăminte sport cu talpă neagresivă pentru sală); materiale sportive specifice ramurilor practicate (mingi, fileu, plase); prezența la activitate în echipament complet este obligatorie.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	Cunoaște la nivel aprofundat principiile fiziologice ale antrenamentului sportiv — supracompensare, periodizare, specificitate, progresivitate — și regulamentele complete ale cel puțin două ramuri sportive practicate; posedă cunoștințe avansate de igienă sportivă, nutriție pentru performanță și metode de recuperare activă și pasivă după efort.
5.2 Aptitudini	Execută cu eficiență tehnică ridicată structuri motrice specifice ramurilor sportive abordate, demonstrând automatizarea deprinderilor consolidate în semestrul I și II; aplică independent metode de antrenament funcțional (HIIT, circuit training, antrenament de forță cu greutatea propriului corp) adaptate obiectivelor personale de fitness; participă activ și constructiv în jocuri sportive de echipă, asumând roluri de jucător și de organizator.
5.3 Responsabilitate și autonomie	Gestionează autonom și responsabil propriul program de activitate fizică pe parcursul semestrului, planificând independent sesiunile de antrenament complementare orelor de curs; promovează în cadrul grupei un comportament fair-play și o cultură a activității fizice regulate; își asumă responsabilitatea pentru propriul nivel de sănătate și condiție fizică, raportând titularului orice modificare a stării de sănătate care poate afecta participarea la activitățile fizice.

6. Conținuturi

6.1 Laborator	Nr. ore	Metode de predare	Observații
Ședința 1: Evaluare inițială a condiției fizice. Testare complexă: alergare 12 minute (testul Cooper), flotări, abdomene, săritură în lungime de pe loc, flexibilitate (aplecarea înainte). Înregistrarea rezultatelor individuale ca punct de referință pentru evaluarea progresului. Stabilirea obiectivelor individuale de condiție fizică pentru semestrul III.	1	Testare practică, individual	
Ședința 2: Antrenament funcțional — circuit training I. Introducere în antrenamentul în circuit (circuit training): principii, beneficii, structura unui circuit. Executarea unui circuit de 6 stații cu exerciții funcționale (flotări, genuflexiuni cu săritură, abdomene V-sit, plank dinamic, burpees, mountain climbers) — 3 runde × 40 secunde lucru / 20 secunde pauză.	1	Activitate practică în circuit	
Ședința 3: Antrenament funcțional — HIIT. Principiile antrenamentului de înaltă intensitate cu intervale (HIIT): raportul lucru/pauză, beneficii cardiovasculare și metabolice. Executarea unui protocol Tabata (8 runde × 20 secunde efort maxim / 10 secunde pauză) alternând exerciții de forță-rezistență. Monitorizarea frecvenței cardiace.	1	Activitate practică HIIT	
Ședința 4: Baschet — tehnica avansată și joc. Recapitulare dribling, pase, aruncare. Elemente tehnice avansate: fenta, pivotul ofensiv, aruncarea din dribling (lay-up), aruncarea de la distanță cu doi pași. Tactică: marcajul om-la-om, apărarea zonă 2-3,	1	Activitate practică, joc bilateral	

contraatacul. Joc bilateral 5×5 cu arbitraj și aplicarea regulamentului complet.			
Ședința 5: Volei — tehnica avansată și joc. Recapitulare serviciu, manșetă, ridicare. Elemente tehnice avansate: serviciul în salt, atacul (smash) cu elan, blocajul individual și colectiv (2 jucători). Tactică: sistemul de joc cu libero, acoperirea atacului. Joc bilateral 6×6 cu arbitraj și aplicarea regulamentului FIVB simplificat.	1	Activitate practică, joc bilateral	
Ședința 6: Fotbal / Futsal — tehnica avansată și joc. Recapitulare pase, conducerea mingii, șut. Elemente tehnice avansate: șutul cu diferite zone ale piciorului, deposedarea, heading. Tactică: presing, contraatacul rapid, marcajul zonal. Joc bilateral futsal 5×5 cu arbitraj. Alternativ (pentru grupe mixte): Ultimate Frisbee — regulament, tehnici de aruncare, tactică de echipă.	1	Activitate practică, joc bilateral	
Ședința 7: Atletism — alergări și sărituri. Tehnica alergării de viteză: startul de jos, faza de accelerare, faza de viteză maximă. Alergare de rezistență: tehnica alergării la tempo uniform, controlul respirației, gestionarea efortului prelungit. Proba de săritură în lungime cu elan: tehnica bătaii, zborului (tehnica 1,5 pași) și aterizării.	1	Activitate practică pe teren	
Ședința 8: Yoga și stretching avansat — recuperare activă. Principiile yoga pentru sportivi: respirația pranayama, Sun Salutation (Surya Namaskar) ca încălzire dinamică. Stretching PNF (Proprioceptive Neuromuscular Facilitation): tehnica contract-relax pentru principalele grupe musculare. Exerciții de mobilitate articulară pentru șold, coloană și umeri. Beneficiile recuperării active.	1	Activitate practică, stretching	
Ședința 9: Antrenament de forță cu greutatea corpului — nivel avansat. Exerciții avansate de forță corporală: flotări cu picioarele sus (decline push-ups), dips la bancă, genuflexiuni pe un picior (pistol squat asistat), extensii pentru spate, plank lateral cu rotație. Construirea unui program săptămânal de forță prin greutatea propriului corp (calisthenics) adaptat obiectivelor individuale.	1	Activitate practică, individual	
Ședința 10: Tenis de masă — tehnica avansată și turneu. Recapitulare serviciu, forehand, backhand. Elemente tehnice avansate: topspin forehand și backhand, blocajul, lob-ul defensiv, efectele pe minge. Tactică: exploatarea unghiurilor, alternarea tempoului, jocul în diagonală vs. pe linie. Mini-turneu cu tablou simplu (eliminare directă sau grupe), arbitraj reciproc.	1	Activitate practică, turneu	

Ședința 11: Dans sportiv și aerobic — coordonare și ritm. Introducere în aerobicul sportiv: principii, structura unei clase de aerobic, tipuri de step. Executarea unui coregrafii de aerobic cu step de intensitate medie (Low Impact Aerobics) de 20 de minute. Elemente de dans sportiv latin (cha-cha-cha, rumba) — pași de bază, mișcarea corpului, coordonarea cu partenera/partenerul.	1	Activitate practică, coregrafie	
Ședința 12: Sport extrem / aventură — escaladă și echilibru. Exerciții de escaladă pe perete artificial (boulder) sau, dacă nu este disponibil, parcurs de obstacole cu exerciții de echilibru, coordonare și forță. Parkour de bază: tehnica rulada, vault-ul de baza (safety vault), echilibrul pe bara de echilibru. Activitate individuală și în perechi cu asigurare reciprocă.	1	Activitate practică individuală	
Ședința 13: Antrenament integrat — concurs sportiv inter-grupe. Organizarea și desfășurarea unui mini-turneu sportiv inter-grupe la sportul ales prin vot (baschet, volei sau futsal). Studenții asigură arbitrajul, organizarea și evidența scorului. Accent pe fair-play, comunicare în echipă și gestionarea competitivității constructive.	1	Concurs sportiv, arbitraj reciproc	
Ședința 14: Evaluare finală a condiției fizice și recapitulare. Retestare cu bateriile de teste din ședința 1: testul Cooper, flotări, abdomene, săritură în lungime de pe loc, flexibilitate. Compararea rezultatelor cu evaluarea inițială și cuantificarea progresului individual. Feedback individual și recomandări pentru menținerea activității fizice pe perioada vacanței.	1	Testare practică, feedback individual	

Bibliografie obligatorie laborator

1. Drăgan, I. (coord.), Medicina sportivă, Ed. Medicală, București, 2002 (retipărit 2018).
2. Bompa, T.O., Buzzichelli, C., Periodization: Theory and Methodology of Training, 6th ed., Human Kinetics, 2019.
3. Regulamentele oficiale FIBA (baschet), FIVB (volei) și FIFA/Futsal — versiunile în vigoare, disponibile la: <https://www.fiba.basketball/> / <https://www.fivb.com/> / <https://www.fifa.com>

Bibliografie complementara laborator

4. McArdle, W.D., Katch, F.I., Katch, V.L., Exercise Physiology: Nutrition, Energy, and Human Performance, 9th ed., Wolters Kluwer, 2022.
5. Stătescu, F., Teoria și metodică educației fizice și sportului, Ed. Pim, Iași, 2010.
6. Ploșteanu, C., Educație fizică și sport — îndrumar metodic, Ed. Universitară Danubius, Galați.

Resurse online gratuite recomandate

7. National Strength and Conditioning Association (NSCA) — resurse de antrenament funcțional și forță: <https://www.nsc.com/>
8. American College of Sports Medicine (ACSM) — ghiduri de activitate fizică și condiție fizică: <https://www.acsm.org/>
9. Darebee — programe gratuite de antrenament funcțional și calisthenics: <https://darebee.com/>

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

Educația Fizică și Sport III continuă și aprofundează traseul de formare motrică și de cultură a activității fizice din semestrul I și II, ridicând nivelul de exigență tehnică și de autonomie a studentului în gestionarea propriului program de activitate fizică. La finalul celor trei semestre de educație fizică, studentul trebuie să fi integrat activitatea fizică regulată ca un comportament de stil de viață, nu doar ca o obligație curriculară — acesta este obiectivul formativ central al disciplinei. Prin introducerea antrenamentului funcțional (circuit training, HIIT, calisthenics), a tehnicilor avansate la sporturile de echipă și a elementelor de yoga și recuperare activă, semestrul III pregătește studenții pentru autogestiunea sănătății fizice pe termen lung, într-un context profesional dominat de sedentarism și de stresul prelungit specific muncii în IT. Aceste competențe răspund direct cerințelor de sănătate ale inginerului TI: prevenirea afecțiunilor musculo-scheletice cauzate de postura prelungită la birou, gestionarea stresului prin activitate fizică și menținerea capacității de concentrare la nivel optim.

Mini-turneul inter-grupe din ședința 13 și evaluarea comparativă a condiției fizice (ședințele 1 și 14) introduc o dimensiune de responsabilitate și de evaluare obiectivă a progresului individual — studentul vede concret, în cifre, rezultatele celor 14 săptămâni de activitate fizică regulată, ceea ce consolidează motivația intrinsecă pentru continuarea practicii sportive dincolo de curriculum.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Nu este cazul — disciplina nu are activitate de curs.	—	—
8.5 Seminar/laborator	Prezența și participarea activă la minimum 10 din 14 ședințe de laborator (75%); progresul demonstrat în condiția fizică generală (compararea rezultatelor testelor finale cu cele inițiale din ședința 1); comportamentul corect în echipă (fair-play, cooperare, respect față de colegi și regulament); prezența în echipament sportiv complet.	Evaluare continuă pe parcursul semestrului: prezență (50%) + progres demonstrat la testul de condiție fizică finală din ședința 14 față de testul inițial din ședința 1 (30%) + comportament și atitudine (fair-play, echipament, implicare) (20%). Nota finală: Admis / Respins	100%
8.6 Standard minim de performanță:			
Prezența la minimum 10 din 14 ședințe de laborator (75%) în echipament sportiv complet ȘI demonstrarea unui progres obiectiv în cel puțin 2 din cele 5 probe ale testului de condiție fizică (Cooper, flotări, abdomene, săritură în lungime de pe loc, flexibilitate) față de evaluarea inițială din ședința 1 — indiferent de nivelul absolut atins. Nu se iau în considerare absențele nemotivate; absențele motivate medical (adeverință) sunt acceptate până la limita a 4 ședințe.			

Data completării 17.03.2026	Semnătura titularului de curs	Semnătura titularului de seminar / laborator Prof. Dr. Ploesteanu Constantin
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	PROGRAMARE LOGICĂ – TIDS219						
1.2 Titularul activităților de curs	Conf. Dr. Bucea Radu Manea Tonis						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Conf. Dr. Bucea Radu Manea Tonis						
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Op DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					19
Documentare					5
Studiu individual					5
Referate					5
Teme casă					2
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Matematici Discrete (TIDS205) — logică propozițională și cu predicate, clauze Horn, rezoluție; Limbaje Formale și Automate (TIDS210) — gramatici formale, limbaje regulate și independente de context; Programare Orientată pe Obiecte (TIDS204) — paradigme de programare, structuri de date.
3.2 de competențe	Cunoașterea logicii cu predicate de ordinul întâi: cuantificatori, unificare, rezoluție; familiarizarea cu cel puțin un limbaj de programare imperativ (C, Java sau Python); înțelegerea conceptelor de recursivitate și structuri de date recursive (liste, arbori).

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă (pentru demonstrații de unificare și rezoluție), videoprojector și acces la internet; platforma e-learning Danubius Online (Sakai); recomandare: instalarea SWI-Prolog înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; SWI-Prolog instalat; SWISH (SWI-Prolog online) ca alternativă; studenții pot folosi propriul laptop; platforma Sakai pentru predarea temelor.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C7. Deține cunoștințe specializate de programare structurată, orientată pe obiecte și funcțională, cu înțelegerea critică a paradigmelor de programare și a domeniilor lor de aplicabilitate. C10. Cunoaște limbajele formale, automatele și principiile translaatoarelor, ca fundament teoretic al proiectării compilatoarelor și al înțelegerii critice a limbajelor de programare moderne.
5.2 Aptitudini	A1. Aplică metode și tehnici avansate de proiectare orientată pe obiecte și utilizează șabloane de proiectare software (design patterns) pentru dezvoltarea de aplicații robuste, reutilizabile și ușor de întreținut.
5.3 Responsabilitate și autonomie	R7. Își asumă rolul de mentor tehnic pentru colegii cu mai puțină experiență, transmițând cunoștințe specializate, oferind feedback constructiv asupra soluțiilor propuse și promovând cultura calității în cadrul echipei.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Paradigme de programare. Prezentare comparativă: programare imperativă, orientată pe obiecte, funcțională, logică. Modelul declarativ: descrierea problemei vs. descrierea soluției. Istoric Prolog: Robinson (1965), Kowalski, Colmerauer. Aplicații ale programării logice: sisteme expert, NLP, baze de date deductive, verificare formală, AI.	2	Prelegere interactivă	
Tema 2: Fundamente Prolog — fapte, reguli, interogări. Structura unui program Prolog: fapte (facts), reguli (rules), interogări (queries). Sintaxa de bază: atomi, variabile, structuri compuse (functori). Baza de cunoștințe (knowledge base). Motorul de inferență: satisfacerea interogărilor prin căutare. Operatorul de conjuncție (,) și disjuncție (;).	2	Prelegere, cod Prolog	
Tema 3: Unificarea. Definiția unificării: cel mai general unificator (MGU). Algoritmul de unificare Robinson. Unificarea în Prolog: operatorul =/2. Diferența dintre unificare și atribuire. Unificarea structurilor compuse. Apariția variabilelor (occur check): ==/2 vs. =/2. Exemplu pas cu pas: urmărirea unificării.	2	Prelegere, demonstrații la tablă	
Tema 4: Mecanismul de căutare și backtracking. Arborele de căutare Prolog: SLD-rezoluție (Selected-Linear-Definite). Ordinea clauzelor în baza de cunoștințe: importanța ordinii. Backtracking automat: când și cum este declanșat. Operatorul cut (!): tăierea ramurii de backtracking, utilizare și pericole. Negația prin eșec (negation as failure): \+/1.	2	Prelegere, trasare arbori	
Tema 5: Structuri de date — liste. Reprezentarea listelor în Prolog: lista ca structură [H T]. Predicatele fundamentale: member/2, length/2, append/3, last/2,	2	Prelegere, cod Prolog	

reverse/2, nth0/3, nth1/3. Implementarea de la zero a predicatelor pe liste cu recursivitate. Acumulatori: reverse eficient cu acumulator.			
Tema 6: Recursivitate și aritmetică. Recursivitate în Prolog: structura clauzei de bază și a clauzei recursive. Calculul recursiv: factorial, Fibonacci, putere. Aritmetica în Prolog: operatorul is/2, operatori de comparare (==, \=, <, >, =<, >=). Diferența dintre = și ==. Recursivitate pe liste cu calcule numerice: sumă, maxim, minim.	2	Prelegere, cod Prolog	
Tema 7: Structuri de date — arbori și grafuri. Reprezentarea arborilor binari în Prolog ca structuri compuse: tree(Val, Left, Right). Operații pe arbori: inserare, căutare, parcurgere (inordine, preordine, postordine). Arbori binari de căutare (BST). Reprezentarea grafurilor: liste de adiacență și predicate de muchie. Parcurgerea grafurilor cu backtracking: DFS în Prolog.	2	Prelegere, cod Prolog	
Tema 8: Controlul execuției și meta-predicatele. Predicatele de control: true/0, fail/0, halt/0. Meta-predicatele: assert/1, retract/1, asserta/1, assertz/1 — baza de cunoștințe dinamică. findall/3, bagof/3, setof/3: colectarea soluțiilor multiple. call/N: apelul dinamic al predicatelor. functor/3, arg/3, =../2 (univ): inspecția structurilor.	2	Prelegere, cod Prolog	
Tema 9: I/O și programare la nivel de sistem. Intrare/ieșire în Prolog: write/1, writeln/1, read/1, nl/0, format/2. Lucrul cu fișiere: open/3, close/1, read_term/2, write_term/2. Consultarea fișierelor: consult/1, use_module/1. Operatorii definiți de utilizator: op/3. Excepțiile: throw/1, catch/3.	2	Prelegere interactivă, cod	
Tema 10: Gramatici de clauze definite (DCG). Motivația DCG: procesarea limbajului natural și parsarea. Sintaxa DCG: reguli de producție ca clauze Prolog. Compilarea DCG în clauze Prolog standard: diferența de liste. Scrierea unui parser DCG pentru expresii aritmetice. Extinderi DCG: condiții extra cu {}/1. Aplicații: analiza sintactică, generarea de propoziții.	2	Prelegere, cod DCG	
Tema 11: Programare cu constrângeri (CLP). Introducere în programarea cu constrângeri logice (CLP): CLP(FD) pentru domenii finite. Biblioteca clpfd în SWI-Prolog: operatorii #=/2, #>/2, #</2, #\=/2. Propagarea constrângerilor vs. generare-și-testare. Etichetarea (labeling). Rezolvarea puzzle-urilor cu CLP(FD): SEND+MORE=MONEY, N-Queens.	2	Prelegere interactivă, cod	
Tema 12: Sisteme expert și raționament bazat pe cunoștințe. Structura unui sistem expert în Prolog: baza de fapte, baza de reguli, motorul	2	Prelegere interactivă, cod	

de inferență. Forward chaining vs. backward chaining. Implementarea unui sistem expert simplu (diagnostic medical sau tehnic). Explicarea raționamentului: trasarea derivărilor. Limite și provocări ale sistemelor expert.			
Tema 13: Prolog și inteligența artificială. Planificarea în Prolog: operatorul STRIPS, generarea planurilor cu backtracking. Rezolvarea problemelor de căutare: A* implementat în Prolog. Procesarea limbajului natural: analiza morfologică și sintactică cu DCG. Prolog și Semantic Web: reprezentarea ontologiilor, interogări SPARQL-like.	2	Prelegere interactivă	
Tema 14: Programare funcțională — fundamente și recapitulare. Paradigma funcțională: funcții ca valori de prim rang, imutabilitate, transparență referențială. Lambda calcul: funcții anonime, aplicare, beta-reducere. Haskell — introducere: tipuri, funcții pure, evaluare leneșă, list comprehensions. Compararea Prolog cu Haskell: similitudini și diferențe față de programarea imperativă. Recapitulare generală.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> 1. Blackburn, P., Bos, J., Striegnitz, K., Learn Prolog Now!, College Publications, 2006. Disponibil gratuit la: https://www.learnprolognow.org/ 2. Bratko, I., Prolog Programming for Artificial Intelligence, 4th ed., Addison-Wesley, 2011. 3. Clocksin, W.F., Mellish, C.S., Programming in Prolog: Using the ISO Standard, 5th ed., Springer, 2003. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> 4. Triska, M., The Power of Prolog — material online gratuit pentru Prolog modern: https://www.metalevel.at/prolog 5. Thompson, S., Haskell: The Craft of Functional Programming, 3rd ed., Addison-Wesley, 2011. <p>Resurse software:</p> <ol style="list-style-type: none"> 6. SWI-Prolog — implementare Prolog gratuită și documentație completă: https://www.swi-prolog.org/ 7. SWISH — SWI-Prolog online, fără instalare: https://swish.swi-prolog.org/ 8. Learn Prolog Now! — versiunea web gratuită: https://www.learnprolognow.org/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Primul program Prolog. Instalarea și configurarea SWI-Prolog. Scrierea primei baze de cunoștințe: fapte despre relații de familie. Interogări simple: membrul, rudenia. Adăugarea de reguli: bunic/2, descendent/2. Urmărirea execuției cu trace/0.	2	Lucrare practică SWI-Prolog	
Lucrarea L2: Unificare și backtracking. Exerciții de urmărire manuală a unificării pentru perechi de termeni dați. Construirea unui arbore de căutare pentru interogări cu mai multe soluții. Experimentarea cu ordinea clauzelor și efectul asupra soluțiilor. Utilizarea cut (!) pentru oprirea backtracking-ului.	2	Lucrare practică SWI-Prolog	
Lucrarea L3: Predicate pe liste — implementare. Implementarea de la zero (fără	2	Lucrare practică SWI-Prolog	

<p>biblioteca) a: member/2, length/2, append/3, last/2, nth/3, delete/3, permutation/2. Testarea completă cu interogări pozitive și negative. Analiza complexității fiecărui predicat.</p>			
<p>Lucrarea L4: Recursivitate și aritmetică. Implementarea recursivă a: factorial/2, fibonacci/2, gcd/3, sum_list/2, max_list/2, flatten/2. Compararea implementării cu acumulator vs. fără acumulator pentru reverse/2: analiza numărului de pași de unificare.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L5: Arbori binari de căutare. Implementarea în Prolog a: insert_bst/3, member_bst/2, inorder/2, build_bst/2 (din listă). Verificarea proprietății BST cu is_bst/1. Calculul înălțimii arborelui. Testarea pe seturi de date variate.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L6: Parcurgerea grafurilor. Reprezentarea unui graf direcționat cu predicate edge/2. Implementarea DFS cu acumulator pentru evitarea ciclurilor: path/3. Găsirea tuturor drumurilor între două noduri cu findall/3. Implementarea BFS cu o coadă reprezentată ca diferență de liste.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L7: Meta-predicate și baza de cunoștințe dinamică. Utilizarea assert/retract pentru construirea dinamică a unei baze de fapte. Implementarea unui sistem de memorizare (memoization) pentru Fibonacci cu assert. Utilizarea findall/3 și bagof/3 pentru colectarea soluțiilor. Implementarea unui predicat maplist/3 de la zero.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L8: Gramatici DCG. Scrierea unui parser DCG pentru propoziții simple în engleză: subiect, predicat, complement. Extinderea cu condiții semantice în {}/1. Scrierea unui parser DCG pentru expresii aritmetice cu priorități. Testarea cu și fără generarea parse tree-ului.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L9: CLP(FD) — probleme combinatoriale. Rezolvarea puzzle-ului SEND+MORE=MONEY cu CLP(FD). Rezolvarea problemei N-Queens pentru N=8 cu CLP(FD): compararea cu implementarea prin backtracking pur. Compararea timpilor de execuție.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L10: Sistem expert simplu. Implementarea unui sistem expert de diagnostic tehnic: fapte despre simptome, reguli de diagnosticare, motorul de inferență backward chaining. Adăugarea explicației: how/1 și why/1 pentru trasarea raționamentului. Extinderea bazei de reguli.</p>	2	Lucrare practică SWI-Prolog	
<p>Lucrarea L11: Rezolvarea puzzle-urilor cu Prolog. Rezolvarea sudoku-ului (4×4) cu CLP(FD) și cu backtracking pur. Rezolvarea</p>	2	Lucrare practică SWI-Prolog	

puzzle-ului turnurilor din Hanoi: generarea secvenței de mutări. Rezolvarea problemei coloritului grafurilor cu constrângeri.			
Lucrarea L12: Planificare și căutare AI. Implementarea algoritmului de planificare cu operatorul STRIPS în Prolog pentru o problemă simplă (lumea blocurilor). Implementarea căutării Best-First cu o euristică simplă. Compararea BFS, DFS și Best-First pe același spațiu de stări.	2	Lucrare practică SWI-Prolog	
Lucrarea L13: Haskell — introducere practică. Primele programe Haskell în GHCi: funcții pure, tipuri de bază, list comprehensions. Implementarea funcțiilor recursive: factorial, fibonacci, quicksort. Funcții de ordin superior: map, filter, foldr, foldl. Compararea stilului Haskell cu stilul Prolog.	2	Lucrare practică GHCi	
Lucrarea L14: Proiect integrat și colocviu. Implementarea în Prolog a unui sistem de rezolvare a unei probleme complexe la alegere: sistem expert, planificator, parser NLP sau rezolvitor de puzzle. Prezentarea bazei de cunoștințe, a interogărilor demo și a arhitecturii soluției. Colocviu oral.	2	Proiect individual, colocviu oral	
	<p>Bibliografie obligatorie laborator</p> <ol style="list-style-type: none"> Blackburn, P., Bos, J., Striegnitz, K., Learn Prolog Now!, College Publications, 2006 — exercițiile practice: https://www.learnprolognow.org/ SWI-Prolog Documentation — referință completă a predicatelor: https://www.swi-prolog.org/pldoc/doc_for?object=manual SWISH — SWI-Prolog online pentru testare rapidă: https://swish.swi-prolog.org/ <p>Bibliografie complementara laborator</p> <ol style="list-style-type: none"> Bratko, I., Prolog Programming for Artificial Intelligence, 4th ed., Addison-Wesley, 2011 — exercițiile de AI. Triska, M., The Power of Prolog — Prolog modern, CLP(FD), DCG: https://www.metalevel.at/prolog <p>Resurse online gratuite recomandate</p> <ol style="list-style-type: none"> SWI-Prolog — implementare gratuită, documentație și tutoriale: https://www.swi-prolog.org/ Learn Prolog Now! — versiunea web gratuită cu exerciții interactive: https://www.learnprolognow.org/ The Power of Prolog (Markus Triska) — resursă avansată gratuită: https://www.metalevel.at/prolog Try Haskell — introducere interactivă în Haskell: https://www.tryhaskell.org/ 		

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

Programarea Logică completează spectrul paradigmelor de programare abordate în program — după programarea imperativă, orientată pe obiecte și funcțională — cu paradigma declarativă, în care programatorul descrie ce este adevărat, nu cum se calculează. Această schimbare de perspectivă dezvoltă un tip de gândire abstractă valoroasă în proiectarea sistemelor complexe, în raționamentul formal și în inteligența artificială, domenii unde Prolog rămâne un instrument activ de cercetare. Componenta de laborator urmărește construirea progresivă a competenței practice: de la primele predicate și mecanismul de backtracking până la sisteme expert, parsere DCG și rezolvitori CLP(FD), culminând cu un proiect integrat la alegere. Introducerea Haskell în Tema 14 și L13 asigură perspectiva comparativă necesară pentru înțelegerea critică a paradigmelor de programare — direct corelată cu C7 din matricea de corelare.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea mecanismelor fundamentale Prolog: unificarea, backtracking-ul, SLD-rezoluția, operatorul cut; scrierea de predicate recursive corecte pe liste și arbori; cunoașterea meta-predicatei (findall, assert, retract); cunoașterea DCG și CLP(FD); înțelegerea comparativă a paradigmelor de programare.	Colocviu scris (1,5 ore): 1 problemă de urmărire manuală a execuției Prolog (arbore de căutare, unificare pas cu pas), 1 problemă de implementare a unui predicat recursiv pe liste sau arbori și 1 problemă de DCG sau CLP(FD) — cu toate clauzele complete și explicarea mecanismului de execuție	60%
8.5 Seminar/laborator	Corectitudinea implementărilor Prolog din laborator (predicate care produc soluțiile corecte pentru toate cazurile de test, inclusiv cazuri limită și backtracking); calitatea proiectului integrat din L14 (baza de cunoștințe structurată, interogări demo relevante); calitatea prezentării și a explicațiilor la colocviul oral din L14.	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L14 (50%)	40%
8.6 Standard minim de performanță:			
Urmărirea corectă și completă la colocviul scris a execuției unui program Prolog dat (arborele de căutare cu toate ramurile, inclusiv cele care eșuează prin backtracking) — aceasta verificând înțelegerea profundă a mecanismului central al limbajului. La colocviul oral din L14: prezentarea unui predicat din proiect cu explicarea pas cu pas a modului în care Prolog găsește prima soluție, demonstrând că studentul înțelege diferența dintre a scrie un predicat și a ști cum îl execută interpretorul.			

Data completării 17.03.2026	Semnătura titularului de curs Conf. Dr. Bucea Radu Manea Tonis	Semnătura titularului de seminar / laborator Conf. Dr. Bucea Radu Manea Tonis
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	PROGRAMARE FUNCȚIONALĂ – TIDS220						
1.2 Titularul activităților de curs	Conf. Dr. Bucea Radu Manea Tonis						
1.3 Titularul activităților de seminar							
1.4 Titularul activităților de laborator	Conf. Dr. Bucea Radu Manea Tonis						
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	C	1.8 Regimul disciplinei	Op DS

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	0/2
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	0/28
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					19
Documentare					5
Studiu individual					5
Referate					5
Teme casă					2
Proiect					
Alte activități (inclusive colocviu, examen)					2
2.9 Total ore pe semestru					75
2.10 Numărul de credite					3

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Matematici Discrete (TIDS205) — logică, inducție, funcții; Limbaje Formale și Automate (TIDS210) — lambda calcul, gramatici; Programare Orientată pe Obiecte (TIDS204) — tipuri, clase, polimorfism; Structuri de Date și Algoritmi (TIDS203) — liste, arbori, recursivitate.
3.2 de competențe	Cunoașterea și utilizarea recursivității în cel puțin un limbaj de programare; înțelegerea noțiunilor de tip, funcție și polimorfism; familiarizarea cu lambda calculul la nivel introductiv; capacitatea de a scrie și depana programe de complexitate medie.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu tablă (pentru demonstrații de tipizare și evaluare), videoprojector și acces la internet; platforma Sakai pentru distribuirea materialelor; recomandare: instalarea GHC (Glasgow Haskell Compiler) și a mediului GHCi înainte de curs.
4.2 de desfășurare a seminarului/laboratorului	Laborator de informatică cu calculatoare individuale; GHC și GHCi instalate; VS Code cu extensia Haskell (HLS) sau Haskell for Mac;

studentii pot folosi propriul laptop; acces la Haskell Playground online pentru testare rapidă.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C7. Deține cunoștințe specializate de programare structurată, orientată pe obiecte și funcțională, cu înțelegerea critică a paradigmatelor de programare și a domeniilor lor de aplicabilitate. C10. Cunoaște limbajele formale, automatele și principiile translaatoarelor, ca fundament teoretic al proiectării compilatoarelor și al înțelegerii critice a limbajelor de programare moderne.
5.2 Aptitudini	A1. Aplică metode și tehnici avansate de proiectare orientată pe obiecte și utilizează șabloane de proiectare software (design patterns) pentru dezvoltarea de aplicații robuste, reutilizabile și ușor de întreținut.
5.3 Responsabilitate și autonomie	R7. Își asumă rolul de mentor tehnic pentru colegii cu mai puțină experiență, transmitând cunoștințe specializate, oferind feedback constructiv asupra soluțiilor propuse și promovând cultura calității în cadrul echipei.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Introducere în programarea funcțională. Caracteristicile paradigmei funcționale: funcții pure, imutabilitate, transparență referențială, evaluare leneșă. Comparatie cu paradigma imperativă și OO. Istoricul Haskell: Lisp, ML, Miranda. De ce Haskell: siguranță la tipizare, conciziune, raționament echațional. Aplicații reale: Pandoc, Xmonad, Facebook Sigma, industria financiară.	2	Prelegere interactivă	
Tema 2: Tipuri și funcții de bază în Haskell. Tipuri primitive: Int, Integer, Float, Double, Bool, Char. Tipul funcție: a -> b. Definiția funcțiilor: ecuații, pattern matching pe valori scalare. Tipuri compuse: tuple (a,b,c), liste [a]. Funcții de bază pe liste: head, tail, length, take, drop, zip, unzip. Tipul String ca [Char].	2	Prelegere, cod GHCi	
Tema 3: Tipizare statică și inferența tipurilor. Sistemul de tipuri Hindley-Milner: tipizare statică puternică. Inferența tipurilor: cum GHC deduce tipul fără adnotații. Polimorfism parametric: funcții generice (id :: a -> a, const :: a -> b -> a). Clase de tipuri (type classes): Eq, Ord, Show, Read, Num, Integral, Fractional. Constrângeri de clasă în semnături de tip.	2	Prelegere, demonstrații GHCi	
Tema 4: Funcții de ordin superior. Funcțiile ca valori de prim rang. Funcții parțial aplicate (currying): f x y = (f x) y. Secțiuni de operatori: (+3), (*2). map :: (a->b) -> [a] -> [b]. filter :: (a->Bool) -> [a] -> [a]. foldr și foldl: reducerea listelor. Compunerea funcțiilor (.) și aplicarea (\$). Stilul point-free.	2	Prelegere, cod Haskell	

Tema 5: List comprehensions și pattern matching avansat. List comprehensions: sintaxă, generatori multipli, garduri. Exemple: mulțimile pythagoreice, sieva lui Eratostene. Pattern matching pe liste: [], (x:xs), (x:y:rest). Pattern matching pe tuple și constructori de date. Wildcards (). Funcții definite prin garduri (guards). Funcția where și let...in.	2	Prelegere, cod Haskell	
Tema 6: Recursivitate și corecție prin inducție. Recursivitate în Haskell: cazul de bază și cazul recursiv. Recursivitate pe liste: sum, product, maximum, reverse, merge sort. Recursivitate pe arbori. Acumulatori: tail recursion. Corecția prin inducție structurală: demonstrarea corectitudinii reverse cu acumulator. Raționamentul echațional: demonstrarea că $\text{map } f . \text{map } g = \text{map } (f.g)$.	2	Prelegere, demonstrații matematice	
Tema 7: Tipuri de date algebrice (ADT). Definirea tipurilor cu data: tipuri produse și tipuri sumă. Constructori de date. Tipuri recursive: lista proprie MyList, arbore binar Tree. Tipuri polimorfice: Maybe a, Either a b. Pattern matching pe constructori. Derivarea claselor: deriving (Show, Eq, Ord). Tipuri newtype.	2	Prelegere, cod Haskell	
Tema 8: Clase de tipuri definite de utilizator. Definirea claselor de tipuri cu class. Instance cu instance. Clasa Functor: fmap. Clasa Foldable: foldr, toList, length, elem. Clasa Traversable. Clase cu superclase. Instance derivate automat. Exemple: clasa Shape cu area și perimenter; clasa Serialize.	2	Prelegere, cod Haskell	
Tema 9: I/O și monada IO. Programele Haskell ca valori pure: separarea efectelor. Tipul IO a: acțiuni I/O. Secvențierea cu do notation: bind (>>=) și then (>>). Funcții I/O de bază: putStr, putStrLn, getLine, readFile, writeFile. Programul main :: IO (). Citirea și parsarea input-ului. Tratarea excepțiilor cu try/catch.	2	Prelegere, cod Haskell	
Tema 10: Monada Maybe și tratarea erorilor. Monada ca pattern de compoziție: return și >>=. Monada Maybe: propagarea erorilor fără excepții. Monada Either: erori cu mesaje. do notation pentru Maybe și Either. Funcțiile fromMaybe, maybe, mapMaybe. Comparație cu null-checking din limbajele imperative. Avantajele tipizării erorilor.	2	Prelegere, cod Haskell	
Tema 11: Evaluare leneșă și fluxuri infinite. Strategia de evaluare leneșă (lazy evaluation): evaluare la nevoie. Liste infinite în Haskell: [1..], repeat, cycle, iterate. Funcțiile take și takeWhile pentru exploatarea listelor infinite. Sieva lui Eratostene ca listă infinită. Fluxuri (streams): corecursivitate. Avantaje și	2	Prelegere interactivă, cod	

dezavantaje ale evaluării leneșe: spațiu, performanță, debugging.			
Tema 12: Parsere funcționale. Tipul Parser a ca funcție String -> [(a, String)]. Constructori de parsere: item, sat, char, digit, letter. Combinatori: (< >), many, many1, sepBy. Parsere pentru expresii aritmetice cu priorități: expr, term, factor. Biblioteca Parsec/Megaparsec: motivație și structură. Comparatie cu parsarea cu yacc/bison.	2	Prelegere interactivă, cod	
Tema 13: Programare concurentă și paralelă în Haskell. Evaluarea paralelă cu par și pseq: strategii de paralelism. Monada STM (Software Transactional Memory): variabile TVar, tranzacții atomice. Fire de execuție ușoare (lightweight threads) cu forkIO. Canale (MVar, Chan). Comparatie cu modelele de concurentă din limbajele imperative.	2	Prelegere interactivă	
Tema 14: Aplicații și recapitulare. Haskell în industrie: Pandoc, Xmonad, Cardano (blockchain). Comparatia paradigmelor: imperativ vs. OO vs. funcțional vs. logic. Haskell vs. Scala vs. F# vs. Clojure: ecosisteme funcționale moderne. Influența Haskell asupra limbajelor mainstream: lambda în Java/Python/C++, LINQ, pattern matching în Rust. Recapitulare generală.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Hutton, G., Programming in Haskell, 2nd ed., Cambridge University Press, 2016. Thompson, S., Haskell: The Craft of Functional Programming, 3rd ed., Addison-Wesley, 2011. Lipovaca, M., Learn You a Haskell for Great Good!, No Starch Press, 2011. Disponibil gratuit la: https://learnyouahaskell.com/ <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Bird, R., Thinking Functionally with Haskell, Cambridge University Press, 2014. O'Sullivan, B., Stewart, D., Goerzen, J., Real World Haskell, O'Reilly Media, 2008. Disponibil gratuit la: https://book.realworldhaskell.org/ <p>Resurse software:</p> <ol style="list-style-type: none"> GHC — compilatorul oficial Haskell și documentație: https://www.haskell.org/ghc/ Haskell.org — resurse oficiale, tutoriale și documentație: https://www.haskell.org/ Hoogle — motorul de căutare pentru funcții Haskell după tip: https://hoogle.haskell.org/ 		
6.2 Laborator	Nr. ore	Metode de predare	Observații
Lucrarea L1: Primul program Haskell. Instalarea GHC și configurarea GHCi. Primele expresii în GHCi: aritmetică, comparații, funcții pe liste. Definirea funcțiilor simple într-un fișier .hs. Utilizarea :type și :info în GHCi. Implementarea funcțiilor: double, square, distance2D.	2	Lucrare practică GHCi	
Lucrarea L2: Tipuri și clase de tipuri. Explorarea sistemului de tipuri cu :type și :kind în GHCi. Scrierea semnăturilor de tip explicite. Testarea polimorfismului: funcții care funcționează pe orice tip Eq, Ord, Num. Implementarea funcțiilor cu constrângeri de clasă: maxim din trei elemente, frecvența unui element într-o listă.	2	Lucrare practică GHCi	

<p>Lucrarea L3: Funcții de ordin superior. Implementarea map, filter, foldr, foldl de la zero. Rezolvarea problemelor cu funcții de ordin superior: sumele pătratelor numerelor impare dintr-o listă, toate cuvintele cu mai mult de n caractere. Compunerea funcțiilor: pipeline-uri cu (.) și (\$) . Scrierea în stil point-free.</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L4: List comprehensions. Rezolvarea problemelor cu list comprehensions: toate perechile (i,j) cu $i < j$ din [1..n], matricea de multiplicare, numerele prime până la n cu sieva, triunghiurile dreptunghice cu perimetru dat. Compararea cu implementarea echivalentă cu map/filter.</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L5: Recursivitate și inducție. Implementarea recursivă: merge sort, quick sort, binary search pe liste sortate. Implementarea cu acumulator: reverse eficient, flatten, concatMap. Demonstrarea echivalenței a două implementări ale aceleiași funcții prin raționament echațional (pe hârtie).</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L6: Tipuri de date algebrice. Definirea unui tip Tree a (Leaf Node a (Tree a) (Tree a)). Implementarea: insert, member, toList (inorder), height, isBST. Definirea tipului Expr pentru expresii aritmetice. Implementarea evaluatorului eval :: Expr -> Int și a afișorului pretty :: Expr -> String.</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L7: Maybe, Either și tratarea erorilor. Implementarea unui calculator sigur cu div și sqrt: rezultate de tip Maybe Double. Parcurgerea unui arbore cu posibile erori: rezultate de tip Either String a. Secvențierea operațiilor cu do notation pentru Maybe. Funcțiile mapMaybe și catMaybes pe liste.</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L8: Clase de tipuri definite de utilizator. Definirea clasei Shape cu metode area :: a -> Double și perimeter :: a -> Double. Instanțe pentru Circle, Rectangle, Triangle. Definirea clasei Describable cu describe :: a -> String. Instanțele derivate automat (Show, Eq). Clasa Functor aplicată tipului Tree.</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L9: I/O și fișiere. Scrierea unui program care citește un fișier text, numără cuvintele și liniile și scrie rezultatul într-un fișier de ieșire. Citirea interactivă: joc simplu de ghicit un număr cu getLine și parsarea input-ului. Gestionarea erorilor I/O cu try/catch.</p>	2	Lucrare Haskell	practică	
<p>Lucrarea L10: Liste infinite și evaluare leneșă. Definirea listelor infinite: fibonacci ca zipWith, primes cu sieva. Funcții pe fluxuri: takeWhile, dropWhile, span. Implementarea unui generator de numere pseudo-aleatoare cu</p>	2	Lucrare Haskell	practică	

iterate. Analiza consumului de memorie: lazy vs. strict evaluation cu seq și \$!.			
Lucrarea L11: Parsere funcționale. Implementarea unui parser minimal pentru expresii aritmetice cu operații +, -, *, / și paranteze. Testarea parserului pe expresii variate. Extinderea parserului cu variabile și un evaluator cu mediu (Map String Int).	2	Lucrare practică Haskell	
Lucrarea L12: Programare concurentă. Implementarea paralelismului de date cu parMap (par + map). Compararea timpilor de execuție: secvențial vs. paralel pe un calcul intensiv (suma pătratelor pe un interval mare). Utilizarea MVar pentru comunicarea între fire. Implementarea unui producer-consumer simplu cu Chan.	2	Lucrare practică Haskell	
Lucrarea L13: Mini-interpreter Haskell. Implementarea unui mini-interpreter pentru un limbaj funcțional simplu: expresii aritmetice, let bindings, funcții anonime (lambda). Tipuri de date: Expr, Value, Env. Funcția eval :: Env -> Expr -> Value. Testarea cu programe simple.	2	Lucrare practică Haskell	
Lucrarea L14: Proiect integrat și colocviu. Implementarea unui proiect la alegere în Haskell: analizor de date CSV, generator de text Markov, rezolvitor de sudoku, mini-compilator. Prezentarea tipurilor definite, a funcțiilor principale și a testelor. Colocviu oral.	2	Proiect individual, colocviu	

Bibliografie obligatorie laborator

1. Hutton, G., Programming in Haskell, 2nd ed., Cambridge University Press, 2016 — exercițiile practice.
2. Lipovaca, M., Learn You a Haskell for Great Good!, No Starch Press, 2011 — versiunea online gratuită: <https://learnyouahaskell.com/>
3. Haskell Documentation — GHC User's Guide și Prelude reference: <https://www.haskell.org/documentation/>

Bibliografie complementara laborator

4. O'Sullivan, B., Stewart, D., Goerzen, J., Real World Haskell, O'Reilly, 2008 — versiunea online gratuită: <https://book.realworldhaskell.org/>
5. Bird, R., Thinking Functionally with Haskell, Cambridge University Press, 2014 — exercițiile de raționament echațional.

Resurse online gratuite recomandate

6. GHC — compilatorul oficial: <https://www.haskell.org/ghc/>
7. Hoogle — căutare funcții Haskell după tip: <https://hoogle.haskell.org/>
8. Learn You a Haskell for Great Good! — tutorial gratuit: <https://learnyouahaskell.com/>
9. Haskell Playground — testare Haskell online: <https://play.haskell.org/>

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

Programarea Funcțională, predată în Haskell, completează peisajul paradigmatelor de programare din program cu abordarea cea mai riguroasă matematic: funcții pure, tipizare statică puternică cu inferență, polimorfism parametric și evaluare leneșă. Aceste concepte nu rămân în sfera academică — ele au migrat masiv în limbajele mainstream: lambda-expresiile din Java 8+, LINQ din C#, pattern matching-ul din Rust, tipurile opționale din Swift și Kotlin sunt toate influențate direct de Haskell și de comunitatea funcțională.

Componenta de laborator urmărește construirea progresivă a competenței: de la primele funcții și tipuri până la parsere, concurență și un proiect integrat. Haskell forțează studentul să gândească în termeni de tipuri înainte de implementare — disciplina tipizării statice puternice formează obiceiul

de a proiecta interfețele unui modul înaintea implementării, direct corelat cu A1 din matricea de corelare.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea și utilizarea tipurilor de bază Haskell și a sistemului de inferență; scrierea funcțiilor de ordin superior (map, filter, foldr); definirea tipurilor algebrice și a claselor de tipuri; cunoașterea monadei IO și Maybe; cunoașterea evaluării leneșe și a listelor infinite; înțelegerea comparativă a paradigmelor de programare.	Colocviu scris (1,5 ore): 1 problemă de derivare a tipului unei expresii Haskell (inferență manuală), 1 problemă de implementare a unei funcții recursive cu tipuri algebrice și 1 problemă de funcții de ordin superior (map/filter/fold) — cu semnăturile de tip complete și explicarea evaluării	60%
8.5 Seminar/laborator	Corectitudinea implementărilor Haskell din laborator (cod care compilează fără erori, producând rezultatele corecte pentru toate cazurile de test); calitatea tipizării (semnături de tip explicite, tipuri bine definite); calitatea proiectului integrat din L14; calitatea prezentării și a explicațiilor la colocviu.	Evaluare continuă: implementări individuale predate pe Sakai după fiecare lucrare (50%) și proiect integrat cu colocviu oral individual la L14 (50%)	40%
8.6 Standard minim de performanță:			
Rezolvarea corectă cu semnăturile de tip complete a cel puțin două din cele trei probleme de la colocviul scris, cu obligativitatea includerii problemei de inferență a tipului — aceasta verificând înțelegerea profundă a sistemului de tipuri Hindley-Milner, fundamentul Haskell. La colocviul oral din L14: explicarea orală a unui tip definit de utilizator din proiect și a motivului pentru care a fost ales (tip produs vs. tip sumă), demonstrând că studentul gândește în tipuri înainte de implementare.			

Data completării 17.03.2026	Semnătura titularului de curs Conf. Dr. Bucea Radu Manea Tonis	Semnătura titularului de seminar / laborator Conf. Dr. Bucea Radu Manea Tonis
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	



UNIVERSITATEA INTERNAȚIONALĂ DANUBIUS
FACULTATEA DE MEDIA, DESIGN ȘI TEHNOLOGIE
DEPARTAMENTUL: INFORMATICĂ ȘI TEHNOLOGII
DOMENIUL DE STUDII DE LICENȚĂ: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
PROGRAMUL DE STUDII: TEHNOLOGIA INFORMAȚIEI
STUDII UNIVERSITARE DE LICENȚĂ CU FRECVENȚĂ
SERIA: 2026-2030
ANUL UNIVERSITAR: 2027-2028

1. Date despre disciplină

1.1 Denumirea disciplinei	PEDAGOGIE II – TIDFAC223						
1.2 Titularul activităților de curs	Lect. Dr. Popa Doinița						
1.3 Titularul activităților de seminar	Lect. Dr. Popa Doinița						
1.4 Titularul activităților de laborator							
1.5 Anul de studiu	II	1.6 Semestrul	1	1.7 Tipul de evaluare	E	1.8 Regimul disciplinei	FAC

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

2.1 Număr de ore pe săptămână	4	din care: 2.2 curs	2	2.3 seminar/laborator	2/0
2.4 Total ore din planul de învățământ	56	din care: 2.5 curs	28	2.6 seminar/laborator	28/0
Distribuția fondului de timp					ore
2.7 Activități monitorizate					-
Practică					-
Elaborare proiect diplomă					-
2.8 Activități individuale					69
Documentare					20
Studiu individual					20
Referate					21
Teme casă					4
Proiect					
Alte activități (inclusive colocviu, examen)					4
2.9 Total ore pe semestru					125
2.10 Numărul de credite					5

3. Precondiții (acolo unde este cazul)

3.1 de curriculum	Psihologia Educației (TIDFAC121) și Pedagogie I (TIDFAC122) — teorii ale învățării, fundamentele procesului de predare-învățare-evaluare, curriculum; Sociologie (TIDC118) — dinamica grupurilor, interacțiunea socială în medii educaționale.
3.2 de competențe	Cunoașterea teoriilor de bază ale învățării (Piaget, Vygotski, Bloom); înțelegerea structurii procesului didactic; capacitatea de a proiecta o secvență simplă de predare; familiarizarea cu principiile evaluării formative.

4. Condiții (acolo unde este cazul)

4.1 de desfășurare a cursului	Sală de curs cu videoproiector, tablă albă și acces la internet; platforma Sakai pentru distribuirea materialelor; studenții vor aduce la curs modele de proiectare didactică și documente curriculare oficiale pentru analiză comparativă.
4.2 de desfășurare a seminarului/laboratorului	Sală de seminar cu acces la internet și videoproiector; activitățile de seminar includ microlecții simulate (microteaching), elaborarea și prezentarea proiectelor didactice, analiza critică a resurselor educaționale digitale și jocuri de rol profesor-elev; studenții filmează opțional microlecțiile pentru autoevaluare.

5. Competențele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei sub forma tabelului de mai jos:

5.1 Cunoștințe	C35. Posedă cunoștințe din domeniul psihologiei educației și al pedagogiei — teorii ale învățării, metode didactice, proiectare curriculară — necesare activității de instruire și transfer de cunoștințe tehnice specializate. C36. Cunoaște principiile didacticii specializării și ale managementului clasei, inclusiv strategii de evaluare formativă și summativă, adaptate predării disciplinelor tehnice din învățământul preuniversitar și universitar.
5.2 Aptitudini	A39. Aplică tehnici pedagogice și didactice adecvate în activitățile de predare, îndrumare și mentoring tehnic, proiectând activități de învățare diferențiată și evaluând progresul cursanților prin metode formative și sumative.
5.3 Responsabilitate și autonomie	R35. Acționează responsabil în rolul de formator sau tutore în activități de instruire asistată de calculator, asigurând calitatea conținutului educațional digital, respectarea drepturilor de autor și accesibilitatea cursurilor pentru toți beneficiarii.

6. Conținuturi

6.1 Curs	Nr. ore	Metode de predare	Observații
Tema 1: Didactica — obiect și statut epistemologic. Didactica generală și didacticile speciale: delimitări conceptuale. Relația didactică: predare, învățare, evaluare ca sistem. Componentele procesului de învățământ: obiective, conținut, metode, mijloace, forme de organizare, evaluare. Paradigme didactice contemporane: constructivismul, didactica cognitivă, didactica socioculturală.	2	Prelegere interactivă	
Tema 2: Proiectarea curriculară. Curriculumul: concept, tipuri (național, local, la decizia școlii). Documentele curriculare: planul de învățământ, programa școlară, manualul. Proiectarea pe competențe: de la competența generală la competența specifică și la activitatea de învățare. Proiectarea didactică anuală, semestrială și pe unitate de învățare.	2	Prelegere interactivă, analiză documente	
Tema 3: Proiectul de lecție. Structura proiectului de lecție: date de identificare, obiective operaționale, conținut, strategia didactică, evaluare. Taxonomia lui Bloom revizuită: niveluri cognitive și formularea obiectivelor operaționale. Tipuri de lecție: de transmitere, de formare, de recapitulare, de evaluare, mixtă. Proiectarea lecției de informatică sau de matematică.	2	Prelegere, modele de proiecte	
Tema 4: Metode didactice — clasificare și utilizare. Metode expositive: prelegerea, explicația, demonstrația. Metode interogative: conversația euristică, problematizarea. Metode activ-participative: studiul de caz, jocul de rol, simularea, proiectul. Metode specifice educației tehnice: experimentul, lucrarea practică, demonstrația cu calculator.	2	Prelegere interactivă	

Alegerea metodei în funcție de obiectiv și de grupul-țintă.			
Tema 5: Mijloace didactice și tehnologie educațională. Clasificarea mijloacelor didactice: intuitive, tehnice, informatice. Tablă interactivă, videoproiector, simulatoare. E-learning și blended learning: platforme LMS (Moodle, Google Classroom, Microsoft Teams). Resurse educaționale deschise (OER). Instruirea asistată de calculator (IAC): tipuri de software educațional. Inteligența artificială în educație: aplicații și limitări etice.	2	Prelegere interactivă	
Tema 6: Evaluarea educațională. Funcțiile evaluării: diagnostică, formativă, sumativă, predicativă. Metode și instrumente de evaluare: probe orale, scrise, practice; portofoliul; proiectul; autoevaluarea; evaluarea prin rubric. Notarea: scale, criterii, descriptori de performanță. Evaluarea competențelor în educația tehnică. Feedback constructiv: principii și tehnici.	2	Prelegere interactivă	
Tema 7: Managementul clasei de elevi. Dimensiunile managementului clasei: ergonomică, psihologică, socială, normativă, operațională, inovatoare. Climatul educațional pozitiv: factori și strategii de creare. Prevenirea și gestionarea comportamentelor disruptive. Motivarea elevilor pentru disciplinele tehnice. Relația profesor-elev: comunicare asertivă, empatie, autoritate.	2	Prelegere interactivă, studii de caz	
Tema 8: Diferențierea și individualizarea instruirii. Stiluri de învățare: modelul VARK. Inteligențele multiple (Gardner): aplicații în proiectarea lecțiilor tehnice. Instruirea diferențiată: strategii pentru elevi cu ritmuri diferite. Educația incluzivă: principii și adaptări curriculare. Elevii supradotați în educația STEM. Bariere în învățarea disciplinelor tehnice și strategii de depășire.	2	Prelegere interactivă	
Tema 9: Proiectarea conținutului digital educațional. Principiile designului instrucțional: modelul ADDIE, SAM. Taxonomia obiectivelor pentru medii digitale (Churches). Multimedia learning (Mayer): principiile reducerii redundanței, coerenței, semnalezării. Elaborarea prezentărilor didactice eficiente. Crearea de videoclipuri educaționale (screencasting). Accesibilitatea conținutului digital: ghidul WCAG.	2	Prelegere interactivă	
Tema 10: E-learning și formarea profesională online. Platforme LMS: Moodle, Canvas, Blackboard, Google Classroom. Tipuri de activități online: lecturi, videoclipuri, forumuri, quiz-uri, teme. Modelul SAMR (Substitution, Augmentation, Modification, Redefinition). Gamificarea în educație:	2	Prelegere interactivă	

puncte, insigne, clasamente. Evaluarea online: anti-plagiat, integritatea academică, e-portfolio. MOOC-uri: oportunități și limitări.			
Tema 11: Mentoring și tutoriat tehnic. Mentoring-ul în formarea profesională: rolul mentorului, etapele relației de mentoring. Tutoriatul în educația universitară: tipuri și tehnici. Feedback eficient: modelul SBI (Situation-Behavior-Impact), feedback sandwich. Coaching-ul educațional. Mentoring-ul peer (coleg-la-coleg) în proiectele software. Aspecte etice ale mentoring-ului: confidențialitate, limite de rol.	2	Prelegere interactivă, studii de caz	
Tema 12: Cercetarea pedagogică aplicată. Tipuri de cercetare în educație: cantitativă, calitativă, acțiune. Cercetarea-acțiune (action research): ciclul plan-act-observe-reflect. Instrumente de cercetare: chestionarul, interviul, observația sistematică, analiza documentelor. Etica cercetării cu subiecți umani. Valorificarea rezultatelor: raportul de cercetare, articolul pedagogic. Baze de date pedagogice: ERIC, Education Source.	2	Prelegere interactivă	
Tema 13: Tendințe în educația tehnică și STEM. Educația STEM/STEAM: abordarea integrată, proiecte interdisciplinare. Computational thinking: fundamente și integrare în curriculum. Robotica educațională și programarea în școală. Makerspaces și fabricare digitală (FabLab) în educație. Învățarea bazată pe proiecte (PBL) în contextul STEM. Viitorul profesiei didactice în era digitală.	2	Prelegere interactivă	
Tema 14: Deontologia profesiei didactice — recapitulare. Statutul profesorului: drepturi, obligații, responsabilități. Codul deontologic al cadrului didactic. Drepturile de autor în educație: utilizarea materialelor third-party, Creative Commons. Protecția datelor elevilor (GDPR în educație). Epuizarea profesională (burnout) în cariera didactică: prevenire și recuperare. Recapitulare generală.	2	Prelegere, recapitulare	
	<p>Bibliografie obligatorie</p> <ol style="list-style-type: none"> Cucoș, C., Pedagogie, ed. a III-a, Ed. Polirom, Iași, 2014. Ionescu, M., Bocoș, M. (coord.), Tratat de didactică modernă, Ed. Paralela 45, Pitești, 2017. Bloom, B.S. et al., Taxonomy of Educational Objectives: The Classification of Educational Goals, Longmans, Green, 1956 — ediția revizuită: Anderson, L.W., Krathwohl, D.R. (eds.), A Taxonomy for Learning, Teaching, and Assessing, Longman, 2001. <p>Bibliografie complementară</p> <ol style="list-style-type: none"> Mayer, R.E., Multimedia Learning, 2nd ed., Cambridge University Press, 2009. Hattie, J., Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement, Routledge, 2009. Joița, E., Instruirea constructivistă — o alternativă: fundamente, strategii, Ed. Aramis, București, 2006. <p>Resurse software:</p> <ol style="list-style-type: none"> Moodle Documentation — ghiduri pentru proiectarea cursurilor online: https://docs.moodle.org/ 		

8. ERIC — baza de date de cercetare pedagogică: <https://eric.ed.gov/>
 9. Creative Commons — ghid pentru utilizarea licențelor CC în educație: <https://creativecommons.org/about/cclicenses/>

6.2 Seminar	Nr. ore	Metode de predare	Observații
S1: Analiza programei școlare. Analiza comparativă a unei programe școlare de informatică sau matematică pentru liceu: identificarea competențelor generale și specifice, a conținuturilor și a activităților de învățare recomandate. Corelarea cu taxonomia Bloom revizuită.	2	Analiză documente, dezbateri	
S2: Elaborarea proiectului didactic I. Elaborarea unui proiect de lecție complet pentru o temă de informatică (ex: algoritmi de sortare, structuri de date, funcții). Formularea obiectivelor operaționale cu verbele taxonomiei Bloom. Prezentarea și feedback în grup.	2	Atelier, peer review	
S3: Elaborarea proiectului didactic II. Proiectarea strategiei didactice: alegerea metodelor, a mijloacelor și a formei de organizare. Elaborarea fișei de lucru pentru elevi. Proiectarea instrumentului de evaluare (grilă de observare sau test scurt). Revizuirea proiectului pe baza feedback-ului.	2	Atelier, peer review	
S4: Microteaching I — prelegerea interactivă. Fiecare student susține o secvență de predare de 5-7 minute pe tema aleasă, în fața grupei (simulare clasă). Feedback structurat pe grilă: claritatea explicațiilor, contactul vizual, gestionarea timpului, utilizarea tablei.	2	Microteaching, feedback pe grilă	
S5: Metode activ-participative în educația tehnică. Aplicarea metodei studiului de caz pe o situație din practică: un bug critic într-un proiect software — echipe de studenți analizează, propun soluții, prezintă în fața clasei. Debrief: cum se adaptează această metodă la elevi de liceu.	2	Studiu de caz, debrief	
S6: Proiectarea evaluării. Elaborarea unui instrument de evaluare sumativă (test de 10 itemi) pentru unitatea de învățare din proiectele didactice. Itemi cu alegere multiplă, itemi cu răspuns scurt, itemi de tip eseu structurat. Elaborarea baremului și a descriptorilor de performanță.	2	Atelier, peer review	
S7: Resurse educaționale digitale. Evaluarea critică a 3 resurse educaționale digitale (videoclip YouTube, quiz Kahoot, lecție Khan Academy) pe baza unei grile cu criterii: corectitudine științifică, accesibilitate, interactivitate, drepturi de autor. Prezentarea concluziilor.	2	Analiză critică, prezentare	
S8: Proiectarea unui curs online simplu. Elaborarea unui modul de curs online în Google Classroom sau Moodle: minim o lecție cu text, un videoclip embed și un quiz de	2	Atelier practic, demonstrație	

autoevaluare cu 5 întrebări. Demonstrarea cursului creat. Feedback pe accesibilitate și claritate.			
S9: Feedback și evaluare formativă. Exersarea tehnicilor de feedback constructiv: modelul SBI aplicat pe proiecte de cod ale colegilor (code review ca evaluare formativă). Analiza unor exemple de feedback ineficient și reformularea lor. Autoevaluarea cu grilă SMART.	2	Joc de rol, exerciții practice	
S10: Diferențierea instruirii. Proiectarea aceluiși conținut (ex: conceptul de recursivitate) pentru trei niveluri de competență diferite: începător, mediu, avansat. Prezentarea și compararea celor trei variante. Debrief: principii de diferențiere în educația tehnică.	2	Atelier, dezbateri	
S11: Microteaching II — lecție cu metode active. Fiecare student susține o secvență de predare de 7-10 minute utilizând o metodă activ-participativă (problematizare, joc de rol sau experiment). Compararea cu Microteaching I: ce s-a îmbunătățit. Feedback pe grilă extinsă.	2	Microteaching, feedback comparativ	
S12: Cercetarea-acțiune în clasă. Proiectarea unui mini-studiu de cercetare-acțiune: identificarea unei probleme din practica predării tehnice, formularea ipotezei, alegerea instrumentului de colectare a datelor (chestionar sau observație). Prezentarea planului.	2	Atelier, prezentare plan	
S13: Deontologie și drepturi de autor în educație. Analiza unor scenarii de dileme etice din practica didactică: utilizarea codului sursă al elevilor fără permisiune, acordarea de note preferențiale, plagiatul. Discutarea soluțiilor etice. Analiza licențelor Creative Commons pentru resurse educaționale.	2	Studii de caz, dezbateri etică	
S14: Portofoliu didactic — prezentare finală. Fiecare student prezintă portofoliul didactic elaborat pe parcursul semestrului: proiect didactic, instrument de evaluare, modul de curs online, reflecție personală. Evaluare pe grilă cu criterii explicite. Feedback final și autoevaluare.	2	Prezentare portofoliu, evaluare finală	

Bibliografie obligatorie seminar

1. Cucuș, C., Pedagogie, ed. a III-a, Ed. Polirom, Iași, 2014 — capitolele de metodologie didactică și evaluare.
2. Ionescu, M., Bocoș, M. (coord.), Tratat de didactică modernă, Ed. Paralela 45, Pitești, 2017 — exercițiile aplicative.
3. Mayer, R.E., Multimedia Learning, 2nd ed., Cambridge University Press, 2009 — designul resurselor digitale.

Bibliografie complementara seminar

4. Hattie, J., Visible Learning: A Synthesis of Over 800 Meta-Analyses Relating to Achievement, Routledge, 2009.
5. Joița, E., Instruirea constructivistă — o alternativă, Ed. Aramis, București, 2006 — activitățile constructiviste.

Resurse online gratuite recomandate

	6. Moodle Documentation — proiectarea cursurilor online: https://docs.moodle.org/ 7. ERIC — baza de date de cercetare pedagogică: https://eric.ed.gov/ 8. Khan Academy — exemple de lecții structurate pentru analiza critică: https://www.khanacademy.org/ 9. Creative Commons — licențe pentru resurse educaționale: https://creativecommons.org/about/cclicenses/
--	--

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

Pedagogie II aprofundează cunoștințele din Pedagogie I prin orientarea spre practica didactică concretă: proiectarea lecției, managementul clasei, evaluarea competențelor și utilizarea tehnologiei educaționale. Disciplina pregătește inginerul TI pentru rolul de formator, tutore sau mentor tehnic — rol tot mai frecvent în industrie (code review, onboarding, training intern) și în învățământul tehnic preuniversitar și universitar.

Structura seminarului, centrată pe microteaching și portofoliul didactic, urmărește formarea competenței practice de predare prin exercițiu direct și feedback structurat — direct corelată cu A39 din matricea de corelare. Componenta de conținut digital educațional și e-learning răspunde cerinței R35: responsabilitatea pentru calitatea și accesibilitatea conținutului creat.

8. Evaluare

Tip activitate	8.1 Criterii de evaluare	8.2 Metode de evaluare	8.3 Pondere din nota finală
8.4 Curs	Cunoașterea principiilor proiectării curriculare și a structurii proiectului de lecție; cunoașterea taxonomiei Bloom revizuite și formularea obiectivelor operaționale; cunoașterea metodelor didactice și a criteriilor de selecție; cunoașterea metodelor și instrumentelor de evaluare formativă și sumativă; cunoașterea principiilor designului instrucțional pentru medii digitale.	Examen scris (1,5 ore): 1 problemă de proiectare didactică (elaborarea obiectivelor operaționale și a strategiei didactice pentru o secvență dată), 1 problemă de evaluare (elaborarea unui item de test cu baremul aferent) și 1 subiect de sinteză (compararea a două metode didactice pentru un context tehnic dat)	50%
8.5 Seminar/laborator	Calitatea portofoliului didactic elaborat pe parcursul semestrului (proiect didactic complet, instrument de evaluare, modul de curs online); calitatea și progresul demonstrat în cele două sesiuni de microteaching (S4 vs. S11); participarea activă la dezbateri și ateliere; calitatea prezentării finale a portofoliului la S14.	Evaluare continuă: portofoliu didactic predat pe Sakai la finalul semestrului (60%) și prezentare finală a portofoliului la S14 cu Q&A de 10 minute (40%)	50%
8.6 Standard minim de performanță:			

Predarea unui portofoliu didactic care conține obligatoriu: un proiect de lecție complet (cu obiective operaționale formulate corect cu verbe din taxonomia Bloom, strategie didactică justificată și instrument de evaluare) ȘI un modul de curs online funcțional cu cel puțin o resursă interactivă (quiz sau activitate). Un proiect de lecție fără obiective operaționale formulate sau fără instrument de evaluare nu îndeplinește standardul minim, indiferent de calitatea celorlalte componente.

Data completării 17.03.2026	Semnătura titularului de curs Lect. Dr. Popa Doinița	Semnătura titularului de seminar / laborator Lect. Dr. Popa Doinița
Data avizării în Consiliul Departamentului 24.03.2026	Semnătura directorului DIT Lect. dr. Antohe Valerian	
Data aprobării în Consiliul Facultății 07.04.2026	Semnătura decanului FMDT Conf. dr. Beteringhe Adrian	