

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Inginerie Software / Master
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	<i>Limbaje si Sisteme de Tipuri</i>			Codul disciplinei	7.20
2.2 Titularul de curs	Prof.dr.ing. Eneia Nicolae Todoran - Eneia.Todoran@cs.utcluj.ro				
2.3 Titularul activităților de seminar / laborator / proiect	Prof.dr.ing. Eneia Nicolae Todoran - Eneia.Todoran@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categorica formativă				DA
	Opționalitate				DOP

3. Timpul total estimate

3.1 Număr de ore pe săptămână	3	din care:	3.2 Curs	2	3.3 Seminar	1	3.3 Laborator	-	3.3 Proiect	-
3.4 Număr de ore pe semestru	42	din care:	3.5 Curs	28	3.6 Seminar	14	3.6 Laborator	-	3.6 Proiect	-
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										15
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										15
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										15
(d) Tutoriat										10
(e) Examinări										3
(f) Alte activități:										
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))					58					
3.9 Total ore pe semestru (3.4+3.8)					100					
3.10 Numărul de credite					4					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Limbaje de Programare, Inginerie Software – cursuri nivel licenta
4.2 de competențe	Operarea cu fundamente stiintifice, ingineresti si matematice

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Interactiune directa (tabla, proiector video, PC cu acces la internet)
5.2. de desfășurare a seminarului / laboratorului / proiectului	Proiector video, tabla, PC cu acces la internet, pentru nota maxima participare activa la minim 70% din seminarii

6. Competențele specifice acumulate

Competențe profesionale	analizeaza specificatii software defineste cerinte tehnice analizeaza grupuri masive de date aplica teoria sistemelor de TIC dezvolta idei creative aliniaza software-ul la arhitcturile de sistem creeaza software efectueza testari de sistem efectueaza cercetare stiintifica furnizeaza documentatie tehnica interpreteaza cerinte tehnice aplica tehnici de analiza statistica supravegheaza dezvoltarea de software utilizeaza biblioteci de software utilizeaza sabloane de proiectare de software
Competențe transversale	Absolventul: • dezvolta o abordare analitica • adopta o abordare proactiva • dezvolta strategii pentru rezolvarea problemelor • este deschis la schimbare • coordoneaza echipele tehnice

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul cunoaste: • algoritmi • modelarea proceselor de afaceri • programarea pe calculator • ciclul de viata al dezvoltarii sistemelor • ingineria sistemelor bazate pe modele • limbajul unificat de modelare (UML) • modelarea orientata pe obiecte • metodologia Agile de management de proiect • metodele de proiectare a sistemelor • formularea algoritmului unui proces • cadre software (framework-uri) • programarea de sisteme TIC • software pentru medii integrate de dezvoltare (IDE) • aplice competente de comunicare in domeniul tehnic
Aptitudini	Studentul este capabil sa: • analizeze specificatii software • defineasca cerinte tehnice • analizeze grupuri masive de date • aplice teoria sistemelor TIC • dezvolte idei creative • alinieze software-ul la arhitecturile de sistem • creeze aplicatii software • supravegheze dezvoltarea de software • dezvolte prototipuri software • utilizeze instrumente de inginerie software asistata de calculator • dezvolte servicii de tip cloud computing • gaseasca solutii pentru probleme

Responsabilitate și autonomie:	Studentul are capacitatea de a lucra independent pentru: • a dezvolta o abordare analitică • a adopta o abordare proactivă • a dezvolta strategii pentru rezolvarea problemelor • a fi deschis la schimbare • a coordona echipele tehnice
--------------------------------	--

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

8.1 Obiectivul general al disciplinei	Obiectivul principal al acestei discipline este de a oferi cunoștințe specifice și de a pregăti studenții în vederea utilizării de modele formale și semantice în proiectarea și verificarea sistemelor de calcul. Cunoștințele sunt prezentate în contextul limbajelor de programare și specificare, cu accent pe verificarea statică a tipurilor, semantica dinamică, modelarea performanței și verificarea formală a proprietăților sistemelor de calcul.
8.2 Obiectivele specifice	Pentru atingerea acestor obiective generale studenții vor: • învăța să specifice și să proiecteze formal limbaje și sisteme de calcul • învăța să verifice formal proprietăți ale limbajelor și sistemelor proiectate • învăța tehnici de proiectare și verificare a proprietăților limbajelor și sistemelor (ex. inducție, semantica de punct fix, bisimulare, coinducție) • învăța să aplice principii și paradigme avansate de proiectare • studia modul în care semantica și modelele formale permit rezolvarea de probleme complexe de proiectare, modelare cantitativă, evaluare performanță • urmări să înțeleagă utilitatea modelelor formale în contextul mai larg al Științei și Calculatoarelor prin exemple (proiectare protocoale, performanță sisteme, modele bazate pe calcul natural, etc.)

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
Introducere, concepte de baza	2	Curs interactiv, expunere, exemple, discutii si intrebari	
Stiluri semantice, semantica operationala	2		
Untyped lambda-calculus (ULC): sintaxa si relatie de evaluare	2		
ULC: termeni lambda De Bruijn (nameless)	2		
Simply typed lambda-calculus (STLC): sintaxa si relatie de tipare	2		
STLC: proprietatile relatiei de tipare	2		
Extensii simple: tuple, recursivitate, liste	2		
Interpretor semantic pentru STLC – implementare Haskell	2		
Introducere in semantica algebrica si semantica de bisimulare	2		
Bisimulare si semantica algebrica: concurenta si nedeterminism	2		
Introducere in CCS (Calculus of Communicating Systems)	2		
Bisimulare si semantica algebrica in CCS (1)	2		
Bisimulare si semantica algebrica in CCS (2)	2		
Proprietati de compozitionalitate	2		
Bibliografie: • B. Pierce, Types and Programming Languages, MIT Press, 2002. • B. Pierce, Advanced Topics in Types and Programming Languages, MIT Press, 2005. • B. Pierce et al, Software Foundations, https://softwarefoundations.cis.upenn.edu, 2025. • R. Milner. Communicating and mobile systems: the pi-calculus, Cambridge Univ. Press, 1999. • D. Sangiorgi, Introduction to Bisimulation and Coinduction, Cambridge University Press, 2011.			

• E.M. Clarke et al, Handbook of Model Checking, Springer, 2018. • R. Harper, Practical Foundations for Programming Languages, Cambridge University Press, 2016. • E.N. Todoran, Types and Programming Languages (lecture notes), Technical University of Cluj-Napoca, https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2025.pdf, 2025. • E.N. Todoran. Introducere in Semantica Limbajelor de Programare, Note de curs si seminar, Universitatea Tehnica din Cluj-Napoca, http://users.utcluj.ro/~eneia/aplc-2016.pdf, 2016.			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Semantica operationala structurata	2	Expunere, rezolvare probleme cu participarea activa a studentilor	
Proiectare semantica cu sisteme de tranzitie	2		
Tehnici de proiectare semantica: semantica directa si de continuare	2		
Termeni lambda De Bruijn (nameless): substitutie si beta-reducere	2		
Proiectare interpretor Haskell pentru STLC	2		
Verificarea sistemelor concurente cu algebre de procese	2		
Demonstrare legi CCS in semantica de bisimulare	2		
Bibliografie			
• B. Pierce, Types and Programming Languages, MIT Press, 2002. • B. Pierce et al, Software Foundations, https://softwarefoundations.cis.upenn.edu, 2025. • R. Milner. Communicating and mobile systems: the pi-calculus, Cambridge Univ. Press, 1999. • E.M. Clarke et al, Handbook of Model Checking, Springer, 2018. • D. Sangiorgi, Introduction to Bisimulation and Coinduction, Cambridge University Press, 2011. • E.N. Todoran, Types and Programming Languages (lecture notes), Technical University of Cluj-Napoca, https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2025.pdf, 2025. • PRISM probabilistic model checker www.prismmodelchecker.org			

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

Aceasta disciplina introduce cunostinte de baza in domeniile Semantica si Metode Formale. Prezentarea este realizata in contextul Limbajelor de Programare si Specificare. Limbajele si sistemele de calcul sunt descrise matematic utilizand sintaxa formala si sunt echipate cu semantica formala in functie de context si necesitate. Din perspectiva ingineriasca, aceasta disciplina este foarte importanta pentru dezvoltarea sistemelor de calcul care impun standarde severe de calitate: fiabilitate, siguranta in functionare, performanta masurabila, etc. Continutul disciplinei este sincronizat cu ultimele progrese in domeniu, pe baza de monografii, studii si cursuri predate la universitati de prestigiu din Europa si SUA.
--

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Abilitatea de rezolvare probleme specifice domeniului. Prezentă, (inter)activitate in timpul orelor de curs	Examen scris – evaluare sumativa	75%
11.5 Seminar/Laborator /Proiect	Abilitatea de rezolvare probleme specifice domeniului. Prezentă, (inter)activitate in timpul orelor de seminar	Elaborare lucrare seminar (articol stiintific), teme seminar saptamanale - evaluare continua	25%
11.6 Standard minim de performanță: nota minima 5 la examenul scris si predarea lucrarii de seminar			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
01.09.2025	Curs	Prof.dr.ing. Eneia Nicolae Todoran	
	Aplicații	Prof.dr.ing. Eneia Nicolae Todoran	

Data avizării în Consiliul Departamentului Calculatoare 17.09.2025	Director Departament Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare 19.09.2025	Decan Prof.dr.ing. Vlad Mureșan