

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 și Tehnologia Informației
1.5 Ciclu de studii	Master
1.6 Programul de studii / Calificarea	Rețele de Comunicații și Sisteme Distribuite / Master
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Algoritmi și Calculabilitate			Codul disciplinei	7.30
2.2 Titularul de curs	Prof. dr. ing. Potolea Rodica - Rodica.Potolea@cs.utcluj.ro				
2.3 Titularul activităților de seminar / laborator / proiect	Prof. dr. ing. Potolea Rodica - Rodica.Potolea@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	1	2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categorica formativă				DS
	Opționalitate				DOP

3. Timpul total estimat

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										40
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										-
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										16
(d) Tutoriat										-
(e) Examinări										2
(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	Structuri de Date, Tehnici de Programare, Algoritmi
4.2 de competențe	Evaluare eficiența algoritmi, Cunoșterea algoritmilor fundamentali pe structurile de date de baza

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator
5.2. de desfășurare a seminarului / laboratorului / proiectului	Calculatoare

6. Competențele specifice acumulate

Competențe profesionale	C1.4 - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor și modelelor de dezvoltare software, precum și a sistemelor software complexe. C1.5 - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate în toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare și integrare, validare). C2 - Elaborarea de tehnici, metode, și metodologii avansate în domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora. C2.1 - Identificarea și descrierea structurii și a modului de funcționare a sistemelor de software complexe și a aplicațiilor dezvoltate pe baza acestora C2.2 - Exploatarea cunoștințelor de specialitate în vederea identificării și înțelegerii metodologiilor și tehnicilor de realizare a componentelor hardware și software C2.3 - Construirea unor componente software originale ale sistemelor avansate de programe, folosind algoritmi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii și medii de programare complexe, raportate în literatura de specialitate. C2.4 - Utilizarea de metode, criterii și metrici de evaluare și selecție a metodologiilor de realizare a sistemelor software, a caracteristicilor lor funcționale și non-funcționale C2.5 - Elaborarea de proiecte software originale, implementarea, testarea și validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate
Competențe transversale	CT3 - Exersarea deprinderii de autoeducare continuă și demonstrarea de abilități critice, inovatoare și de cercetare

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul cunoaște: - programare pe calculator - software pentru mediu integrat de dezvoltare
Aptitudini	Studentul este capabil să: - definească cerințele tehnice - analizeze specificații software - dezvolte prototipul pentru software - remedieze erorile din software - utilizeze biblioteci de software
Responsabilitate și autonomie:	Studentul are capacitatea de a lucra independent pentru: - a dezvolta o abordare analitică - a da dovadă de dorință de învățare - a dezvolta strategii pentru rezolvarea problemelor - a fi deschis la schimbare - a lucra eficient în echipe

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	Analiza obiectiva a unei probleme și identificarea complexității sale; identificarea de soluții potențiale și alegerea celei potrivite contextului dat.
---------------------------------------	---

8.2 Obiectivele specifice	Calculul de complexitate; diferentierea claselor de complexitate; cunoșterea tipurilor de probleme care nu au soluții „usoare”.
---------------------------	---

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
Complexitate	2	Expunere la tablă, prezentare cu videoproiectorul, discuții	
Timp polinomial, P si NP	2		
NP completitudine	2		
Reducibilitate	2		
P=?NP	2		
Demonstrare NP-comp	2		
Probleme NP-comp	2		
Model Computational	2		
Masina Turing	2		
Reducere NP-comp1 (Circuit-SAT, SAT, 3-FNC-SAT)	2		
Reducere NP-comp2 (Clica, Acoperire varfuri, Suma)	2		
Reducere NP-comp3 (Ciclu Hamiltonian, Comis Voiajor)	2		
Aplicatii NP-comp si aproximari 1	2		
Aplicatii NP-comp si aproximari 2	2		
Bibliografie: 1. Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. Introduction to Algorithms . 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN: 9780262032933. 2. Papadimitriou, C. H. Computational Complexity . 1st ed. Boston: Addison Wesley Publishing Company, 1994. ISBN: 0201530821. 3. Arora, Sanjeev and Barak, Boaz <i>Complexity Theory: A Modern Approach</i> , Princeton University, available on the web, http://www.cs.princeton.edu/theory/index.php/Compbook/Draft 4. Sipser, Michael. <i>Introduction to the Theory of Computation</i> . 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972.			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
P vs NPC	1	Expuneri la tablă, explicații suplimentare, discuții.	Nu este cazul
Demonstrari apartenenta NPC	1		
Algoritmi de reducere1	1		
Algoritmi de reducere2	1		
Aproximari1	1		
Aproximari2	1		
Probleme NPC	1		
Bibliografie: [1] Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. Introduction to Algorithms . 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN: 9780262032933 [2] Papadimitriou, C. H. Computational Complexity . 1st ed. Boston: Addison Wesley Publishing Company, 1994. ISBN: 0201530821 [3] Arora, Sanjeev and Barak, Boaz <i>Complexity Theory: A Modern Approach</i> , Princeton University, available on the web, http://www.cs.princeton.edu/theory/index.php/Compbook/Draft [4] Sipser, Michael. <i>Introduction to the Theory of Computation</i> . 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972			

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

Identificarea problemelor “dificile” și rezolvarea lor aproximativă; cunoașterea de soluții exacte ale problemelor “usoare”.
--

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	Examinare	Examen scris	50%
11.5 Seminar /Laborator /Proiect	Prezentare/eseu/rezolvare probleme complexe	Prezentare (P)	50%
Standard minim de performanță: Min 5 (50%E+50%P>=5)			

Data completării: 01.02.2025	Titulari	Titlu Prenume NUME	Semnătura
	Curs	Prof.dr.ing. Rodica POTOLEA	
	Aplicații	Prof.dr.ing. Rodica POTOLEA	

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