

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
1.8 Codul disciplinei	7.1

2. Date despre disciplină

2.1 Denumirea disciplinei	Statistica si probabilitati				
2.2 Titularii de curs	Prof. Dr. Ioan Rasa, Ioan.Rasa@math.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof. Dr. Ioan Rasa, Ioan.Rasa@math.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	1	Laborator	-	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	14	Laborator	-	Proiect	-
3.3 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:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										58
3.5 Total ore pe semestru (3.2+3.4)										100
3.6 Numărul de credite										4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Analiza matematica, algebra liniara, matematici speciale
4.2 de competențe	Competentele disciplinelor de mai sus

5. Condiții (acolo unde este cazul)

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

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C1 - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> • C1.1 - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software • C1.2 - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate • C1.3 - Utilizarea unor modele și metode specifice pentru identificarea de componente și soluții software viabile în condiții de specificare
-----------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>parțială</p> <ul style="list-style-type: none"> • 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). <p>C2 - Elaborarea de tehnici, metode, și metodologii avansate în domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> • 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
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Înșuirea principiilor fundamentale ale teoriei probabilităților și statisticii matematice
7.2 Obiectivele specifice	Capacitatea de a calcula probabilități, de a modela în limbaj probabilistic probleme cu caracter practic și de a folosi metodele statisticii în situații concrete

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Probabilitate, entropie, informație	2		
Variabile aleatoare discrete	2		
Variabile aleatoare continue	2		
Media și dispersia	2		
Metoda celor mai mici pătrate	2		
Corelație și regresie	2		
Lanturi Markov	2		
Distributia limita. Exemple	2		
Lanturi Markov ascunse	2		
Testarea ipotezelor statistice	2		
Tehnici Bayes de estimare	2		
Familii Gaussiene	2		
Metoda verosimilității maxime	2		
Algoritmul EM	2		
<p>Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)</p> <p>1. Ioan Rasa, Lectures on Probability Theory and Stochastic Processes, U.T.Pres 2006</p> <p>2. Ioan Rasa, Teoria Probabilităților și Aplicații, ITCN 1994</p>			

3. C.Jalobeanu, I.Rasa, Incertitudine si decizie. Statistica si probabilitati aplicate in management, U.T.Pres 2001			
4. T.K.Moon, Wynn C.Stirling, Mathematical Methods and Algorithms for Signal Processing, Prentice Hall 2000.			
5. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, Wiley-Interscience, 2004			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Probabilitate, entropie, informatie	1		
Variabile aleatoare discrete, Variabile aleatoare continue	1		
Media si dispersia, Metoda celor mai mici patrate	1		
Lanturi Markov	1		
Tehnici Bayes de estimare	1		
Familii Gaussiene	1		
Algoritmul EM	1		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
1. Ioan Rasa, Lectures on Probability Theory and Stochastic Processes, U.T.Pres 2006			
2. Ioan Rasa, Teoria Probabilitatilor si Aplicatii, ITCN 1994			
3. C.Jalobeanu, I.Rasa, Incertitudine si decizie. Statistica si probabilitati aplicate in management, U.T.Pres 2001			
4. T.K.Moon, Wynn C.Stirling, Mathematical Methods and Algorithms for Signal Processing, Prentice Hall 2000.			
5. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, Wiley-Interscience, 2004			

*Se vor preciza, după caz: tematica seminarilor, lucrările de laborator, tematica și etapele proiectului.

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

Continutul disciplinei a fost discutat cu colegii din alte departamente in scopul coroborarii cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Capacitatea de a prezenta un subiect teoretic cu demonstratii	Lucrare scrisa (teorie)	25%
Seminar	Abilitatea de a rezolva probleme specifice domeniului	Lucrare scrisa (probleme)	75%
Laborator			
Proiect			
Standard minim de performanță: Nota 5.			

Titularul de Disciplina
Prof.dr.Ioan Rasa

Director departament
Prof.dr.ing. Rodica Potolea

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
1.8 Codul disciplinei	7.2

2. Date despre disciplină

2.1 Denumirea disciplinei	Limbaje si Sisteme de Tipuri				
2.2 Titularii de curs	Prof.dr.ing. Enea Todoran – Enea.Todoran@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof.dr.ing. Enea Todoran – Enea.Todoran@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	1	Laborator	-	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	14	Laborator	-	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										25
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										10
(d) Tutoriat										10
(e) Examinări										3
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										58
3.5 Total ore pe semestru (3.2+3.4)										100
3.6 Numărul de credite										4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Limbaje de programare (curs nivel licența)
4.2 de competențe	Operarea cu fundamente științifice, ingineresti și matematice

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator (prezentare interactivă); pentru nota maximă, prezenta la curs minim 70%
5.2. de desfășurare a seminarului / laboratorului / proiectului	Pentru nota maximă, prezenta la seminar minim 90% cu participarea activă

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C1 - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> • C1.1 - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software • C1.2 - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate • C1.3 - Utilizarea unor modele și metode specifice pentru identificarea
-----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>de componente si solutii software viabile în condiții de specificare parțială</p> <ul style="list-style-type: none"> • C1.4 - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor si modelelor de dezvoltare software, precum si a sistemelor software complexe. • C1.5 - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate in toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare si integrare, validare). <p>C2 - Elaborarea de tehnici, metode, si metodologii avansate in domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> • 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 algoritimi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii si medii de programare complexe, raportate in literatura de specialitate. • C2.4 - Utilizarea de metode, criterii si metrice de evaluare si selectie a metodologiilor de realizarea a sistemelor software, a caracteristicilor lor funcționale și non-funcționale • C2.5 - Elaborarea de proiecte software originale, implementarea, testarea si validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Obiectivul principal al acestei discipline este de a oferi cunostinte specifice si de a pregati studentii in vederea utilizarii de modele formale si semantice in proiectarea si verificarea sistemelor de calcul. Cunosintele sunt prezentate in contextul limbajelor de programare si specificare, cu accent pe verificarea statica a tipurilor, semantica dinamica, modelarea performantei si verificarea formala a proprietatilor sistemelor de calcul.
7.2 Obiectivele specifice	<p>Pentru atingerea acestor obiective generale studentii vor:</p> <ul style="list-style-type: none"> • Invata sa specifice si sa proiecteze formal limbaje si sisteme de calcul • Invata sa verifice formal proprietati ale limbajelor si sistemelor proiectate • Invata tehnici de proiectare si verificare a proprietatilor limbajelor si sistemelor (ex. inductie, semantica de punct fix, bisimulare, coinductie) • Invata sa aplice principii si paradigme avansate de proiectare • Studia modul in care semantica si modelele formale permit rezolvarea de probleme complexe de proiectare, modelare cantitativa, evaluare performanta • Urmari sa inteleaga utilitatea modelelor formale in contextul mai larg al Stiintei si Calculatoarelor prin exemple (proiectare protocoale, performanta sisteme, modele bazate pe calcul natural, etc.)

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
----------	--------	-------------------	------------

Introducere, concepte de baza	2	Expunere la tabla, prezentare cu videoproiector, discutii	Nu este cazul
Semantica operationala	2		
Semantica denotationala	2		
Recursivitate si semantica de punct fix	2		
Stare, control, evaluare	2		
Tehnici semantice (continuari, monade)	2		
Domenii semantice	2		
Nedeterminism si concurenta	2		
Bisimulare, semantica algebrica	2		
Semantica de continuare pentru calcul distribuit	2		
Semantica statica, verificare tipuri	2		
Subiecte avansate: calcul distribuit si global	2		
Subiecte avansate: modelare performanta	2		
Subiecte avansate: calcul natural	2		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
1. J.W. De Bakker, E.P. De Vink. <i>Control flow semantics</i> . MIT Press, 1996.			
2. J. Hillston, <i>A compositional approach to performance modeling</i> , Cambridge University Press, 1996.			
3. J. Hillston, <i>Performance modeling</i> , http://www.inf.ed.ac.uk/teaching/courses/pm/ , lecture notes, 2011.			
4. B. Jacobs, J. Rutten, An introduction to (co)algebras and (co)induction, In D. Sangiorgi, J. Rutten, editors, <i>Advanced topics in bisimulation and coinduction</i> , pp. 38-99, http://homepages.cwi.nl/~janr/papers/files-of-papers/2011_Jacobs_Rutten_new.pdf , 2011			
5. R. Milner. <i>Communicating and mobile systems: the pi-calculus</i> . Cambridge Univ. Press, 1999.			
6. B. Pierce, (Ed.). <i>Advanced topics in programming languages and type systems</i> . MIT Press, 2005.			
7. B. Pierce. <i>Programming languages and type systems</i> . MIT Press, 2002.			
8. A. Pitts. <i>Denotational semantics</i> , http://www.cl.cam.ac.uk/teaching/1112/DenotSem/dens-notes-bw.pdf , lecture notes, 2012.			
9. F. Turbak, D. Gifford. <i>Design concepts in programming languages</i> . MIT Press, 2009.			
10. E.N. Todoran. <i>Limbaje si sisteme de tipuri</i> . Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, http://users.utcluj.ro/~eneia/fl.htm , 2010.			
11. E.N. Todoran. <i>Limbaje si sisteme de tipuri – semantica denotationala</i> . Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, http://users.utcluj.ro/~eneia/fl.htm , 2012.			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	
Semantica operationala structurata	1	Expuneri la tabla, explicatii suplimentare, discutii, rezolvare de probleme cu participarea activa a studentilor	Nu este cazul
Proiectare cu sisteme de tranzitie	1		
Semantica denotationala	1		
Domenii semantice	1		
Semantica statica, verificare tipuri	1		
Studiu de caz (calcul distribuit / calcul natural)	1		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
12. J.W. De Bakker, E.P. De Vink. <i>Control flow semantics</i> . MIT Press, 1996.			
13. J. Hillston, <i>A compositional approach to performance modeling</i> , Cambridge University Press, 1996.			
14. J. Hillston, <i>Performance modeling</i> , http://www.inf.ed.ac.uk/teaching/courses/pm/ , lecture notes, 2011.			
15. B. Jacobs, J. Rutten, An introduction to (co)algebras and (co)induction, In D. Sangiorgi, J. Rutten, editors, <i>Advanced topics in bisimulation and coinduction</i> , pp. 38-99, http://homepages.cwi.nl/~janr/papers/files-of-papers/2011_Jacobs_Rutten_new.pdf , 2011			
16. R. Milner. <i>Communicating and mobile systems: the pi-calculus</i> . Cambridge Univ. Press, 1999.			
17. B. Pierce, (Ed.). <i>Advanced topics in programming languages and type systems</i> . MIT Press, 2005.			
18. B. Pierce. <i>Programming languages and type systems</i> . MIT Press, 2002.			
19. A. Pitts. <i>Denotational semantics</i> , http://www.cl.cam.ac.uk/teaching/1112/DenotSem/dens-notes-bw.pdf , lecture notes, 2012.			
20. F. Turbak, D. Gifford. <i>Design concepts in programming languages</i> . MIT Press, 2009.			
21. E.N. Todoran. <i>Limbaje si sisteme de tipuri</i> . Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, http://users.utcluj.ro/~eneia/fl.htm , 2010.			
22. E.N. Todoran. <i>Limbaje si sisteme de tipuri – semantica denotationala</i> . Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, http://users.utcluj.ro/~eneia/fl.htm , 2012.			

**Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

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

Această disciplină introduce cunoștințe de bază în domeniile Semantica și Metode Formale. Prezentarea este realizată în contextul Limbajelor de Programare și Specificare. Limbajele și sistemele de calcul sunt descrise matematic utilizând sintaxa formală și sunt echipate cu semantica formală în funcție de context și necesitate. Din perspectiva ingineriască, această disciplină este foarte importantă pentru dezvoltarea sistemelor de calcul care impun standarde severe de calitate: fiabilitate, siguranța în funcționare, performanța măsurabilă, etc. Conținutul disciplinei este sincronizat cu ultimele avansuri în domeniu, pe baza de monografii, studii și cursuri predate la universități de prestigiu din Europa și SUA. Disciplina a fost evaluată odată cu programul de studiu master în Ingineria Calculatoarelor de către ARACIS.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea de rezolvare a unor probleme specifice domeniului. Prezentă, (inter)activitate în timpul orelor de curs	Examen scris	70%
Seminar	Abilitatea de rezolvare a unor probleme specifice domeniului. Prezentă, (inter)activitate în timpul orelor de seminar	Elaborare paper științific	30%
Laborator			
Proiect			

Standard minim de performanță:
Modelarea și rezolvarea unor probleme de proiectare semantica limbaje sau sisteme, utilizând aparatul formal specific domeniului.

Titularul de Disciplină
Prof.dr.ing. Eneia Todoran

Director departament
Prof.dr.ing. Rodica Potolea

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
1.8 Codul disciplinei	7.3

2. Date despre disciplină

2.1 Denumirea disciplinei	Algoritmi si Calculabilitate				
2.2 Titularii de curs	Prof. dr. ing. Rodica Potolea, Rodica.Potolea@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof. dr. ing. Rodica Potolea, Rodica.Potolea@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	1	Laborator	-	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	14	Laborator	-	Proiect	-
3.3 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:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))									58	
3.5 Total ore pe semestru (3.2+3.4)									100	
3.6 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 eficienta algoritmi, Cunosterea 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

6.1 Competențe profesionale	<p>C1 - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> • C1.1 - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software • C1.2 - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate • C1.3 - Utilizarea unor modele și metode specifice pentru identificarea
-----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>de componente si solutii software viabile în condiții de specificare parțială</p> <ul style="list-style-type: none"> • C1.4 - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor si modelelor de dezvoltare software, precum si a sistemelor software complexe. • C1.5 - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate in toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare si integrare, validare). <p>C2 - Elaborarea de tehnici, metode, si metodologii avansate in domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> • 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 algoritimi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii si medii de programare complexe, raportate in literatura de specialitate. • C2.4 - Utilizarea de metode, criterii si metrice de evaluare si selectie a metodologiilor de realizarea a sistemelor software, a caracteristicilor lor funcționale și non-funcționale • C2.5 - Elaborarea de proiecte software originale, implementarea, testarea si validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate
6.2 Competențe transversale	CT3 - Exersarea deprinderii de autoeducare continuă și demonstrarea de abilități critice, inovatoare și de cercetare

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Analiza obiectiva a unei probleme si identificarea complexitatii sale; identificarea de solutii potientiale si alegerea celei potrivite contextului dat.
7.2 Obiectivele specifice	Calcul de complexitate; diferentierea claselor de complexitate; cunosterea tipurilor de probleme care nu au solutii „usoare”.

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Complexitate	2		
Timp polinomial, P si NP	2		
NP completitudine	2		
Reductibilitate	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 (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			

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 *Complexity Theory: A Modern Approach, Princeton University, available on the web*, <http://www.cs.princeton.edu/theory/index.php/Compbook/Draft>
4. Sipser, Michael. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972

8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
P vs NPC	1		
Demonstrari apartenenta NPC	1		
Algoritmi de reducere1	1		
Algoritmi de reducere2	1		
Aproximari1	1		
Aproximari2	1		
Probleme NPC	1		

Bibliografie (*bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător*)

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 *Complexity Theory: A Modern Approach, Princeton University, available on the web*, <http://www.cs.princeton.edu/theory/index.php/Compbook/Draft>
4. Sipser, Michael. *Introduction to the Theory of Computation*. 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

9. 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"

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Examinare	Examen scris (E)	80%
Seminar	Prezentare/eseu/rezolvare probleme complexe	Prezentare (P)	20%
Laborator			
Proiect			

Standard minim de performanță:
Min 5 (80%E+20%P>=5)

Titularul de Disciplina
Prof. dr. ing. Rodica Potolea

Director departament
Prof. dr. ing. Rodica Potolea

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
1.8 Codul disciplinei	8.

2. Date despre disciplină

2.1 Denumirea disciplinei	Sisteme interactive				
2.2 Titularii de curs	Prof.dr.ing. Dorian Gorgan, dorian.gorgan@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof.dr.ing. Dorian Gorgan, dorian.gorgan@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DA
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	1	Laborator	-	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	14	Laborator	-	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										21
(d) Tutoriat										4
(e) Examinări										3
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										58
3.5 Total ore pe semestru (3.2+3.4)										100
3.6 Numărul de credite										4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea într-un limbaj obiectual de nivel înalt (Ex. C++, Java).
4.2 de competențe	Metodologia de dezvoltare a unei aplicații software.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Prezență la curs minim 75% pentru admiterea la examenul final
5.2. de desfășurare a seminarului / laboratorului / proiectului	Prezență obligatorie 100% pentru admiterea la examenul final

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C1 - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineriei și informatice avansate</p> <ul style="list-style-type: none"> • C1.1 - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software • C1.2 - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate <p>C1.3 - Utilizarea unor modele și metode specifice pentru</p>
-----------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>identificarea de componente si solutii software viabile în condiții de specificare parțială</p> <ul style="list-style-type: none"> • C1.4 - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor si modelelor de dezvoltare software, precum si a sistemelor software complexe. • C1.5 - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate in toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare si integrare, validare). <p>C3 - Specificarea, analiza, modelarea, proiectarea, verificarea, testarea, validarea, si mentenanta sistemelor software avansate și a componentelor software, folosind instrumentele specifice domeniului</p> <ul style="list-style-type: none"> • C3.1 - Demonstrarea cunoașterii tehnologiilor, mediilor de programare, a uneltelor CASE de dezvoltare software și a conceptelor sistemelor de programe complexe • C3.2 - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de realizare a sistemelor software complexe -propușe în literatura științifică • C3.3 - Analiza, modelarea și proiectarea inovativă a sistemelor de calcul și a aplicațiilor informatice, a componentelor hardware și software aferente • 3.4 - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate • C3.5 - Dezvoltarea și implementarea de soluții software originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Principalul obiectiv al disciplinei este dezvoltarea aplicațiilor interactive prin însușirea tehnicilor de analiză, specificare, proiectare, implementare și evaluare a componentelor care asigură interacțiunea cu utilizatorul. Se evidențiază conceptele și tehnicile din ingineria software specifice metodologiilor orientate utilizator.
7.2 Obiectivele specifice	<p>Pentru atingerea acestor obiective generale, studenții vor învăța să:</p> <ul style="list-style-type: none"> • Proiecteze arhitectura sistemelor software interactive. • Utilizeze unelte software pentru dezvoltarea aplicațiilor interactive; • Desfășoare o activitate de cercetare bibliografică și experimentală, ale carei rezultate sunt redactate într-o lucrare științifică; • Realizeze o sinteză și o analiză științifică și, de a susține o prezentare orală a unei teme științifice; • Realizeze un proiect conform metodologiei de dezvoltare și evaluare a aplicațiilor interactive orientate utilizator; • Lucreze individual sau în echipa.

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere. Istoric.	2	Expunere la tablă, prezentare cu videoproietorul, discuții	Nu este cazul
Conceptele dezvoltării interfețelor utilizator.	2		
Conceptele de comunicare intrări și ieșiri.	2		
Proiectarea orientată utilizator.	2		
Metodologia proiectării interfețelor utilizator.	2		
Utilizabilitatea în interfețele utilizator grafice.	2		
Definirea cerințelor utilizator. Descrierea și analiza taskurilor.	2		

Prototipizarea interfeței utilizator. Evaluarea interfeței utilizator.			
Tehnici și stiluri de interacțiune.	2		
Tehnici de interacțiune cu obiecte din spațiul virtual.	2		
Tehnici de interacțiune cu suprafețe modelate prin particule .	2		
Interfețe multimodale. Subiecte din cercetarea științifică actuală.	2		
Modele de obiecte active.	2		
Tehnologii utilizate in interfețele utilizator actuale: tehnologii Web, servicii Web, Web semantic, terminale wireless, tehnologii multimedia.	2		
Unelte, medii de lucru si limbaje utilizate pentru dezvoltarea interfețelor utilizator grafice.	2		
Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>) In biblioteca UTC-N 1. B. Shneiderman, Designing the User Interface. Strategies for Effective Human Computer Interaction, Addison-Wesley, 1992. 2. A. Watt, F. Policarpo, 3D Games. Real-time Rendering and Software Technology, Addison-Wesley, 2001. In bibliotecile virtuale 1. Curs Sisteme Interactive, http://cgis.utcluj.ro/education/71-is 2. Resurse curs Sisteme Interactive, http://cgis.utcluj.ro/didactic			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Conceptele dezvoltării interfețelor utilizator.	1	Studii de caz pe subiecte din domeniul aplicațiilor interactive, exemplificări prin utilizarea uneltelor software și a tehnologiilor specializate, expuneri la tablă, explicații suplimentare, discuții.	Nu este cazul.
Proiectarea orientată utilizator. Metodologia proiectării interfețelor utilizator.	1		
Definirea cerințelor utilizator. Descrierea și analiza taskurilor. Prototipizarea interfeței utilizator. Evaluarea interfeței utilizator.	1		
Tehnici și stiluri de interacțiune. Tehnici de interacțiune cu obiecte din spațiul virtual.	1		
Interfețe multimodale. Subiecte din cercetarea științifică actuală.	1		
Tehnologii utilizate in interfețele utilizator actuale: tehnologii Web, servicii Web, Web semantic, terminale wireless, tehnologii multimedia.	1		
Unelte, medii de lucru si limbaje utilizate pentru dezvoltarea interfețelor utilizator.	1		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>) In biblioteca UTC-N 1. B. Shneiderman, Designing the User Interface. Strategies for Effective Human Computer Interaction, Addison-Wesley, 1992. 2. A. Watt, F. Policarpo, 3D Games. Real-time Rendering and Software Technology, Addison-Wesley, 2001.			

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Spre deosebire de ingineria software consacrată, cursul prezintă metodologia orientată utilizator, folosită pe scară largă în dezvoltarea aplicațiilor interactive. Se studiază și exemplifică tehnici specifice acestei metodologii bazate pe concepul de utilizabilitate, scenariul utilizator, prototipizare, metafore, evaluare cognitivă, evaluare euristică, interacțiune multimodală etc. Conținutul disciplinei a fost discutat cu actori importanți din acest domeniu, atât din mediul academic cât și cel industrial, din România sau alte țări. Disciplina a fost evaluată de către ARACIS, odată cu alte programe de studiu de master.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	La examenul scris (E) se verifică însușirea cunoștințelor predate la curs. Activitatea la curs (AC) reflectă participarea activă la prezentările și dezbaterile științifice de la curs.	Examen scris Verificări pe parcurs, discuții	40% 10%
Seminar	Lucrarea științifică (L) demonstrează capacitatea de realizare a unui studiu științific și elaborarea unei lucrări sau raport științific. Proiectul (P) demonstrează abilitatea utilizării metodologiei de dezvoltare a aplicațiilor interactive.	Lucrare științifică, Proiect	50%
Laborator			
Proiect			
Standard minim de performanță: Nota finală: $N = 0,4 * E + 0,5 * (L+P) / 2 + 0,1 * AC$ Condiția de obținere a creditelor: $N \geq 5$; $E \geq 5$; $L \geq 5$; $P \geq 5$; $AC \geq 5$.			

Titularul de Disciplina
Prof. dr. ing. Dorian Gorgan

Director departament
Prof.dr.ing. Rodica Potolea

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
1.8 Codul disciplinei	9.

2. Date despre disciplină

2.1 Denumirea disciplinei	Managementul Proiectelor				
2.2 Titularii de curs	Prof.dr.ing. Mihaela Dinsoreanu- Mihaela.Dinsoreanu@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof.dr.ing. Mihaela Dinsoreanu- Mihaela.Dinsoreanu@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DA
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	-	Laborator	1	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	-	Laborator	14	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										20
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										10
(d) Tutoriat										5
(e) Examinări										3
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										58
3.5 Total ore pe semestru (3.2+3.4)										100
3.6 Numărul de credite										4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Inginerie software
4.2 de competențe	Elaborarea de tehnici, tehnologii, metode și metodologii specifice sistemelor informatice

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Prezență la curs minim 50% pentru admiterea la examenul final
5.2. de desfășurare a seminarului / laboratorului / proiectului	Prezență obligatorie 100% pentru admiterea la examenul final

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C3 - Specificarea, analiza, modelarea, proiectarea, verificarea, testarea, validarea, si mentenanta sistemelor software avansate și a componentelor software, folosind instrumentele specifice domeniului</p> <ul style="list-style-type: none"> • C3.1 - Demonstrarea cunoașterii tehnologiilor, mediilor de programare, a uneltelor CASE de dezvoltare software și a conceptelor sistemelor de programe complexe • C3.2 - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de realizare a sistemelor software complexe -propane
-----------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>în literatura științifică</p> <ul style="list-style-type: none"> • C3.3 - Analiza, modelarea și proiectarea inovativă a sistemelor de calcul și a aplicațiilor informatice, a componentelor hardware și software aferente • C3.4 - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate • C3.5 - Dezvoltarea și implementarea de soluții software originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate <p>C4 - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> • C4.1 - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module • C4.2 - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații • C4.3 - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe • C4.4 - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe. • C4.5 - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat.
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	
7.2 Obiectivele specifice	<p>Pentru atingerea acestor obiective generale, studenții vor:</p> <ul style="list-style-type: none"> • Urmări să înțeleagă conceptele componente ale unui plan de proiect precum și tehnicile de elaborare a planului de proiect. • Studia tehnici și unelte specifice diferitelor aspecte ale gestionării proiectelor (ex. gestionarea pachetelor de lucru, timpului, resurselor, costului etc) • Studia tehnici și unelte de estimare, monitorizare și control, analiza a riscurilor etc. • Învăța să analizeze specificațiile și să elaboreze un plan de proiect; • Învăța să monitorizeze și controleze evoluția proiectului până la finalizare aplicând metode specifice;

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere	2	Expunere la tablă, prezentare cu videoprojectorul, discuții	
Disciplina Managementului de Proiect – Noțiuni de baza	2		
Managementul Proiectelor în Metodologiile Agile	2		
Managementul Proiectelor în Procesul Unificat	2		
Rolul Managerului de Proiect	2		
Planificarea și Adaptarea Procesului	2		
Planificarea Disciplinelor	2		
Planificarea și Adaptarea Proiectului	2		
Planificarea Rolurilor de Proiect	2		
Planificarea anvergurii, WBS și planificarea în timp	2		

Planificarea Resurselor	2		
Planificarea mediului si managementul schimbarii	2		
Managementul Riscului	2		
Estimarea si Esalonarea Efortului	2		
Bibliografie			
1. <i>The Unified Software Development Process (Hardcover)</i> Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 1998.			
2. <i>Object Solutions : Managing the Object-Oriented Project (Addison-Wesley Object Technology Series)</i> , Grady Booch, Addison Wesley, 1995			
3. <i>Software Project Management: A Unified Framework</i> , Walker Royce, Addison Wesley			
4. <i>Planning Extreme Programming</i> , Kent Beck, Addison Wesley, 2000			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Planificarea si Adaptarea Procesului	1	Rezolvarea unor exercitii, explicații suplimentare discuții	
Planificarea Disciplinelor	1		
Planificarea si Adaptarea Proiectului	1		
Planificarea Rolurilor de Proiect	1		
Planificarea anvergurii, WBS si planificarea in timp	1		
Planificarea Resurselor	1		
Planificarea mediului si managementul schimbarii	1		
Bibliografie			
1. <i>The Unified Software Development Process (Hardcover)</i> Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 1998.			
2. <i>Object Solutions : Managing the Object-Oriented Project (Addison-Wesley Object Technology Series)</i> , Grady Booch, Addison Wesley, 1995			
3. <i>Software Project Management: A Unified Framework</i> , Walker Royce, Addison Wesley			
4. <i>Planning Extreme Programming</i> , Kent Beck, Addison Wesley, 2000			

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

Întrucât această disciplină este foarte importantă pentru gestionarea proiectelor software, conținutul ei este cât se poate de modern deoarece enunță principiile gestionării proiectelor software și prezintă apoi tehnici de gestionare a tuturor aspectelor implicate. Conținutul disciplinei a fost discutat cu actori importanți din acest domeniu, atât academici cât și industriali, din România, Europa și S.U.A. Disciplina a fost evaluată, o dată cu programul de studiu de master Inginerie Software, de către ARACIS.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de curs	Examen scris	60%
Seminar			
Laborator	Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de aplicatii	Rezolvarea unor probleme specifice	40%
Proiect			
10.4 Standard minim de performanta: Planificarea si monitorizarea proiectelor software, utilizând aparatul formal specific domeniului.			

Titularul de Disciplina
Prof.dr.ing Mihaela Dinsoreanu

Director departament
Prof.dr.ing. Rodica Potolea

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
1.8 Codul disciplinei	10.

2. Date despre disciplină

2.1 Denumirea disciplinei	Proiectare software sensibil la context				
2.2 Titularii de curs	Conf. Dr. ing. Tudor Cioara tudor.cioara@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Conf. Dr. Ing. Ionut Anghel ionut.anghel@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	-	Laborator	1	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	-	Laborator	14	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										10
(d) Tutoriat										16
(e) Examinări										2
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										58
3.5 Total ore pe semestru (3.2+3.4)										100
3.6 Numărul de credite										4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	n/a
4.2 de competențe	Operarea cu fundamente științifice, ingineresti și ale informaticii

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Prezență la curs minim 50% pentru admiterea la examenul final.
5.2. de desfășurare a seminarului / laboratorului / proiectului	Prezență obligatorie 100% pentru admiterea la examenul final.

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C4 - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> • C4.1 - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate si integrare specifice sistemelor software, luate atât în ansamblu cât și pe module • C4.2 - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații • C4.3 - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul
-----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<p>complexe</p> <ul style="list-style-type: none"> • C4.4 - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe. • C4.5 - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat. <p>C5 - Îmbinarea creativă a cunoștințelor multidisciplinare din domeniul științei calculatoarelor și tehnologiei informației în vederea cercetării, specificării, proiectării, optimizării, implementării, testării și evaluării de teorii, algoritmi, tehnici, metode și metodologii originale specifice sistemelor software complexe.</p> <ul style="list-style-type: none"> • C5.1 - Demonstrarea cunoașterii metodologiei de cercetare, proiectare, implementare, optimizare și testare a sistemelor de calcul complexe • C5.2 - Demonstrarea capacității de a analiza și interpreta situații noi prin prisma cunoștințelor fundamentale din domeniul calculatoarelor și tehnologiei informației • C5.3 - Îmbinarea creativă, bazată pe descoperirea de legături semantice și funcționale noi, a diferite principii de proiectare moderne din domeniul calculatoarelor și tehnologiei informației pentru rezolvarea unor probleme de optimizare • C5.4 - Fundamentarea activității de cercetare și proiectare inovativă din domeniul calculatoarelor pe criterii corecte de evaluare • C5.5 - Realizarea de activități de cercetare cu finalitate practică demonstrată prin prototipuri software și / sau hardware funcționale.
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Principalul obiectiv al acestei discipline este de introduce concepte de baza și tehnologii specifice integrării și utilizării stream-urilor mari de date ce sunt achiziționate din surse de date eterogene (dispozitive IoT, web, social networking, etc.) în aplicații software. Astfel se urmărește asigurarea bagajului de cunoștințe ce poate fi folosit la evaluarea critică a soluțiilor de dezvoltare existente, precum și la dezvoltarea și integrarea de noi soluții în aplicații existente sau nou dezvoltate.
7.2 Obiectivele specifice	Pentru atingerea obiectivului general, studenții vor studia arhitectura Lambda și tehnologiile specifice pentru fiecare element al arhitecturii: modele de date, managementul eficient al fluxurilor de date și procesarea lor atât offline cât și în real time pentru a oferi diferite vederi utilizatorilor aplicațiilor web.

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere, overview curs, motivație, probleme legate procesarea datelor în timp real	2	Expunere la tablă, prezentare cu video-proiectorul, discuții	Nu este cazul.
Arhitecturi, Cerințe nonfuncționale, limbaje de programare	2		
Modele de date scalabile	2		
Stări distribuite, Consens și Coordonare	2		
Managementul Fluxurilor de Date Distribuite	2		
Cozi și fluxuri de date	2		
Stocarea datelor distribuite	2		
Batch Layer: Map Reduce	2		
Apache Spark Streaming	2		
Apache Spark Machine Learning	2		

Serving Layer	2		
Speed Layer – vederi in timp real	2		
Speed Layer – procesarea streamurilor de micro – batch-urilor	2		
Concluzii, directii de dezvoltare in cercetare	2		
Bibliografie			
1. Byron Ellis, Real Time Analytics Techniques to Analyze and Visualize Streaming Data-Wiley (2014)			
2. Nathan Marz and James Warren, Big Data, Principles and best practices of scalable realtime data systems, April 2015, ISBN 9781617290343, Manning Publications			
3. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia - Learning Spark Lightning-Fast Big Data Analysis-O'Reilly Media			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Mapare teme studenti, Discutarea temelor de prezentare primite de studenți (utilizarea tehnologiilor si uneltelor software existente in proiectarea si implementarea aplicatiilor bazate pe procesarea stream-urilor de date)	1	Expuneri la tablă, prezentare cu video-proiectorul, explicații suplimentare, discuții.	Nu este cazul.
Arhitectura Lambda si Apache Trift	1		
Apache ZooKeeper, Apache Kafka, Apache Flume	1		
Hadoop file system, MongoDB si Cassandra	1		
Apache Spark Batch layer si Serving Layer	1		
Speed Layer - Micro Batch Stream	1		
Finalizare seminar, discutii proiectare aplicatii distribuite complexe	1		
Bibliografie			
1. Byron Ellis, Real Time Analytics Techniques to Analyze and Visualize Streaming Data-Wiley (2014)			
2. Nathan Marz and James Warren, Big Data, Principles and best practices of scalable realtime data systems, April 2015, ISBN 9781617290343, Manning Publications			
3. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia - Learning Spark Lightning-Fast Big Data Analysis-O'Reilly Media			

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

In cadrul studiilor de licenta, studentii aprofundeaza proiectarea a multor clase de sisteme printre care se numara sistemele distribuite, sistemele paralele etc. Aceasta disciplina isi propune sa completeze varietatea de sisteme acoperite in timpul studiilor de licenta, propunand studierea unei clase de sisteme distribuite la scala mare si aplicatii software complexe care devine din ce in ce mai prezenta atat in aria de cercetare cat si in domeniul comercial.

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitatea de a propune solutii unor probleme specifice domeniului. Prezență, (inter)activitate în timpul orelor de curs.	Examen scris.	60%
Seminar			
Laborator	Abilitatea de a identifica probleme și de a utiliza tehnologiile existente în domeniu in dezvoltarea de aplicatii web complexe. Prezență, (inter)activitate în timpul orelor.	Prezentari articole si proiect.	40%
Proiect			
Standard minim de performanta: Înțelegerea conceptelor de baza ale domeniului și demonstrarea abilității de a utiliza noile tehnologii studiate.			

Titularul de Disciplina
Conf. Dr. Ing. Tudor Cioara

Director departament
Prof.dr.ing. Rodica Potolea

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	Inginerie Software/ Master
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	11.

2. Date despre disciplină

2.1 Denumirea disciplinei	Proiect IS 2				
2.2 Titularii de curs	Prof. dr. ing. Ioan Salomie - Ioan.Salomie@cs.utcluj.ro				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Prof.dr.ing.Gorgan Dorian - Dorian.Gorgan@cs.utcluj.ro Prof.dr.ing. Potolea Rodica- Potolea.Rodica@cs.utcluj.ro Prof.dr.ing. Eneia Todoran – Eneia.Todoran@cs.utcluj.ro Conf.dr.ing. Mihaela Dinsoreanu - Mihaela.Dinsoreanu@cs.utcluj.ro				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (E – examen, C – colocviu, V – verificare)	C
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	-	din care:	Curs	-	Seminar	-	Laborator	-	Proiect	2
3.2 Număr de ore pe semestru	-	din care:	Curs	-	Seminar	-	Laborator	-	Proiect	28
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										20
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										20
(d) Tutoriat										10
(e) Examinări										2
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										72
3.5 Total ore pe semestru (3.2+3.4)										100
3.6 Numărul de credite										4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Proiect IS 1
4.2 de competențe	Aferente disciplinei de mai sus

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Nu este cazul
5.2. de desfășurare a seminarului / laboratorului / proiectului	Echipe și programe specifice temei de proiect

6. Competențele specifice acumulate

6.1 Competențe profesionale	C4 - Integrarea contextuală și integritatea sistemelor software complexe <ul style="list-style-type: none"> • C4.1 - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module • C4.2 - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații
-----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<ul style="list-style-type: none"> • C4.3 - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe • C4.4 - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe • C4.5 - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Dezvoltarea de competente și abilități pentru elaborarea de proiecte din domeniul ingineriei software, calculatoarelor și tehnologiei informației
7.2 Obiectivele specifice	Asimilarea de cunoștințe și abilități privind: <ul style="list-style-type: none"> • elaborarea arhitecturii unei aplicații din domeniul ingineriei software • proiectarea componentelor de bază ale aplicației • elaborarea documentației de proiectare

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Nu e cazul	-		
Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
-			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Introducere	2	Prezentare metodologii de proiectare, Verificări periodice	
Elaborarea arhitecturii: - Identificarea principalelor componente	2		
Elaborarea arhitecturii: - Identificarea și stabilirea interfețelor	2		
Elaborarea arhitecturii: - Identificarea și stabilirea aspectelor dinamice	2		
Prezentarea arhitecturii elaborate (document)	2		
Proiectare: - Detalierea componentelor arhitecturale (1)	2		
Proiectare: - Detalierea componentelor arhitecturale (2)	2		
Proiectare: - Detalierea componentelor arhitecturale (3)	2		
Proiectare: - Detalierea componentelor arhitecturale (4)	2		
Proiectare: - Detalierea componentelor arhitecturale (5)	2		
Proiectare: - Detalierea componentelor arhitecturale (6)	2		
Proiectare: - Detalierea componentelor arhitecturale (7)	2		
Prezentarea proiectului (document)	2		
Elaborarea și prezentarea documentației finale	2		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
[1] R.S. Pressman - Software Engineering, A Practitioner's Approach, McGraw-Hill, 7/e, 2009			
[2] Ian Sommerville – Software Engineerig, Addison Wesley, 8/e (2006), 9/e (2011)			
[3] Eric Braude, Michael Bernstein, Software Engineering – Modern Approaches, Wiley 2010			
[4] Capers Jones - Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, McGraw-Hill, 2010			
[5] Peter Eeles, Peter Cripps - The Process of Software Architecting, Addison Wesley, 2010			

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

--

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Nu este cazul		
Seminar			
Laborator			
Proiect	Pe baza rezultatelor practice si a referatului elaborat	Evaluare orala Evaluare referat	60% 40%
Standard minim de performanță: Nota 5			

Titularul de Disciplina
Prof.dr.ing.Ioan Salomie

Director departament
Prof.dr.ing. Rodica Potolea

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
1.8 Codul disciplinei	12.

2. Date despre disciplină

2.1 Denumirea disciplinei	Activitate de cercetare 2				
2.2 Titularii de curs	Nu e cazul				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Nu e cazul				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare (<i>E – examen, C – colocviu, V – verificare</i>)	V
2.7 Regimul disciplinei	<i>DA – de aprofundare, DS – de sinteza, DC – complementară</i>				DS
	<i>DI – Impusă, DOp – opțională, DFac – facultativă</i>				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	-	din care:	Curs	-	Seminar	-	Laborator	-	Proiect	14
3.2 Număr de ore pe semestru	-	din care:	Curs	-	Seminar	-	Laborator	-	Proiect	196
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										50
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										
(d) Tutoriat										
(e) Examinări										4
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))										54
3.5 Total ore pe semestru (3.2+3.4)										250
3.6 Numărul de credite										10

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Activitatea de cercetare 1
4.2 de competențe	Competențele disciplinei de mai sus

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Nu este cazul
5.2. de desfășurare a seminarului / laboratorului / proiectului	Echipamente si programe specifice temei de proiect

6. Competențele specifice acumulate

6.1 Competențe profesionale	<p>C3 - Specificarea, analiza, modelarea, proiectarea, verificarea, testarea, validarea, si mentenanta sistemelor software avansate și a componentelor software, folosind instrumentele specifice domeniului</p> <ul style="list-style-type: none"> • C3.1 - Demonstrarea cunoașterii tehnologiilor, mediilor de programare, a uneltelor CASE de dezvoltare software și a conceptelor sistemelor de programe complexe • C3.2 - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de realizare a sistemelor software complexe -propane în literatura științifică
-----------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<ul style="list-style-type: none"> • C3.3 - Analiza, modelarea și proiectarea inovativă a sistemelor de calcul și a aplicațiilor informatice, a componentelor hardware și software aferente • C3.4 - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate • C3.5 - Dezvoltarea și implementarea de soluții software originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate <p>C4 - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> • C4.1 - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module • C4.2 - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații • C4.3 - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe • C4.4 - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe • C4.5 - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat
6.2 Competențe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Dezvoltarea de abilitați și competente de cercetare și proiectare în domeniul ingineriei software, calculatoarelor și tehnologiei informațiilor
7.2 Obiectivele specifice	<p>Asimilarea de cunoștințe și abilitați privind:</p> <ul style="list-style-type: none"> • elaborarea schemei generale sau a arhitecturii sistemului software ce urmează a fi dezvoltat • efectuarea de experimente, teste și verificări • enunțarea unor ipoteze de lucru și validarea acestora prin experimente • proiectarea componentelor unui sistem aplicativ

8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Nu e cazul.			
Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)			
-			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Definirea obiectivelor activității de cercetare pe care o va realiza în lucrarea de dizertație; Stabilește programul de cercetare teoretică, experimentală și/sau prin simulare numerică pe care îl va realiza în lucrarea de dizertație; Documentare asupra temei de dizertație; Realizarea unui raport de sinteză a activităților derulate.		Lucru individual și verificări periodice	
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>) Se stabilește de către fiecare îndrumător de proiect de dizertație în parte.			

**Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

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

--

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Nu este cazul		
Seminar			
Laborator			
Proiect	Pe baza rezultatelor practice si a referatului elaborat	Evaluare orala Evaluare referat	60% 40%
Standard minim de performanță: Nota 5			

Titularul de Disciplina
Indrumatorii de disertație

Director departament
Prof. dr. ing. Rodica Potolea