



FIȘA DISCIPLINEI

(licență)

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Vasile Alecsandri” din Bacău
1.2. Facultatea	Facultatea de Inginerie
1.3. Departamentul	Departamentul de Energetică și Știința Calculatoarelor
1.4. Domeniul de studii	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii	Licență
1.6. Programul de studii/calificarea	Tehnologia Informației
1.7. Forma de învățământ	Învățământ cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Sisteme tolerante la defecte				
2.2. Titularul activităților de curs	Conf. dr. ing. Popescu Cornel				
2.3. Titularul activităților de seminar	Conf. dr. ing. Popescu Cornel				
2.4. Anul de studiu	IV	2.5. Semestrul	07	2.6. Tipul de evaluare	E
2.7. Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DD - în domeniu, DS - de specialitate, DC - complementară				DS
	Categorია de opționalitate a disciplinei: DI - obligatorie (impusă), DO - opțională (la alegere), DL - facultativă (liber aleasă)				DI

3. Timpul total estimat (ore alocate activităților didactice)

3.1. Număr de ore pe săptămână	3	3.2. Curs	2	3.3. Seminar/Laborator/Proiect	0/1/0
3.4. Totalul de ore pe semestru din planul de învățământ	42	3.5. Curs	28	3.6. Seminar/Laborator/Proiect	0/14/0
Distribuția fondului de timp pe semestru:					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Tutoriat					14
Examinări					2
Alte activități (precizați): referat					-

3.7. Total ore studiu individual	58
3.8. Total ore pe semestru	100
3.9. Numărul de credite	4

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Inginerie software (nivel mediu) și programare orientată pe obiecte-POO
4.2. de competențe	Arhitectura sistemelor de calcul.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	• Sala cu proiector
5.2. de desfășurare a seminarului/laboratorului/proiectului	• Sala calculatoare legate la internet. Tabla, videoprojector, calculator, software specific disciplinei

6. Competențe specifice acumulate

6.1. Competențe profesionale	<p>C5.1. Identificarea și descrierea instrumentelor de modelare, simulare și evaluare a performanțelor sistemelor hardware, software și de comunicații</p> <p>C5.2. Utilizarea unor cunoștințe interdisciplinare pentru asigurarea exploatării sistemelor hardware, software și de comunicații în raport cu cerințele domeniului de aplicații</p> <p>C5.3. Utilizarea unor principii și metode de bază pentru asigurarea securității, siguranței și ușurinței în exploatare a sistemelor hardware, software și de comunicații</p> <p>C5.4. Testarea și evaluarea calitativă a caracteristicilor funcționale și nefuncționale ale sistemelor informatice, pe baza unor criterii specifice</p> <p>C5.5. Dezvoltarea de sisteme și aplicații pentru întreținerea și utilizarea de sisteme hardware, software și de comunicații</p>
6.2. Competențe transversale	

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

7.1. Obiectivul general al disciplinei	Însușirea principalelor tehnici de proiectare pentru sistemele tolerante la defectari, utilizarea de metode și tehnici de testare a aplicațiilor software și a sistemelor software embedded, cunoașterea modelelor și metodelor utilizate în analiza și designul sistemelor tolerante la defecte și a sistemelor cu fiabilitate ridicată, deoarece toleranța la defecte constituie un obiectiv primordial în implementarea sistemelor specializate pe aplicații precum rețelele de comunicații și controlul zborului, familiarizarea cu conceptele de bază și state-of-the-art conexe analizei și designului sistemelor tolerante la defecte, prin studierea sistemelor tolerante la defecte existente, împreună cu tehnicile aplicate, iar aspectele practice ale toleranței la defectare vor fi observate prin intermediul lucrărilor practice de laborator.
7.2. Obiectivele specifice	<p>Competențe specifice (vizează competențele asigurate de programul de studiu);</p> <ul style="list-style-type: none"> ➤ Stăpânirea noțiunilor teoretice, privind STD, aplicarea elementelor teoretice și practice. ➤ Eficientizarea proiectării mecanismelor de tolerare a defectelor ➤ Prezentarea conceptelor generale, a elementelor fundamentale privind metodele, tehnicile și procedurile de testare a produselor software (în special cele embeded), precum și a standardelor din domeniu, relationându-se cu exemple pentru dezvoltări de tip ciclu în V și cu studii de caz. ➤ Însușirea fundamentelor și a unor metode de testare, analiză și verificare a aplicațiilor software, scrierea de teste și folosirea de utilitare de testare. ➤ Dezvoltarea de competențe manageriale legate de planificarea, gestionarea activităților și a resurselor specifice, precum și analiza rezultatelor procesului de testare.
(Cunoaștere și înțelegere)	<ul style="list-style-type: none"> • Cunoașterea și utilizarea adecvată a noțiunilor specifice domeniului STD;
Explicare și interpretare,	<ul style="list-style-type: none"> • Familiarizarea cu sistemele actuale, cu tehnica și cu echipamentele uzuale; • Explicarea și înțelegerea conceptelor de toleranță la defecte;
Instrumental – aplicative	<ul style="list-style-type: none"> • Disponibilitate totală în lucrul cu sistemele informatice; • Capacitate sporită de învățare intuitivă, bazată pe analogii, exemple diverse și similitudini; • Dezvoltarea aptitudinilor de operare cu noțiuni specifice; preluarea și implementarea cu ajutorul acestora, a unor aspecte ale realității în cadrul unor aplicații formale.
Atitudinale	<ul style="list-style-type: none"> • Manifestarea unor atitudini pozitive și responsabile față de domeniul științific și tehnic și valorificarea optimă și creativă a propriului potențial în activitățile științifice și tehnice; • Implicarea în promovarea și dezvoltarea inovațiilor științifice și tehnice; • Participarea la propria dezvoltare profesională și științifică. • Creativitate și pragmatism în ingineria STD

8. Conținuturi

	Curs	Nr. ore	Metode de predare	Observații
01	Principii generale privind sistemele tolerante la defecte. (1) Structura sistemului și implementarea toleranței la defecte. (2) Modelarea conceptului de toleranță la defecte. (3) Strategii de implementare a toleranței la defecte. (4) Perspective privind dezvoltarea sistemelor tolerante la defecte.	2		
02	Concepte și definiții fundamentale caracteristice domeniului testării. Diagnosticitate. Strategii de implementare a toleranței la defecte Fiabilitatea sistemelor de calcul (1) Scopuri ale toleranței la defectare. (2) Definiții (3) Aplicații ale sistemelor tolerante la defectare; (a) Sisteme cu	2	Prelegere, discuții	

	functionare îndelungata (b) Aplicatii critice de calcul (c) Aplicatii necesitând: Amânarea mentenantei și Disponibilitate ridicata. (4) Toleranta la defectare ca obiectiv de proiectare (a) Definitii (b) Cauzele si caracteristicile defectelor		Prezentări interactive.	
03	Utilizarea redundanței ca abordare a toleranței la defecte (1) Redundanța fizică; (a) Replicare active (b) Backup primar. (2) Redundanța informațională; (a) Codificarea datelor (b) Coduri de paritate (c) Sume de verificare (d) Coduri M-din-N (e) Coduri ciclice de codificare (f) Coduri aritmetice	2	Interacțiune a directă prin forumul / blogul disponibil al site-ului cursului.	
04	Analiza unor tehnici de implementare a toleranței la defecte. Tehnici de detectie a erorilor (1) Considerații critice asupra tehnicilor de implementare a toleranței la defecte. (2) Tehnici de reconfigurarea a sistemelor la apariția defectărilor. (3) Model al reconfigurării în sistemele multiprocesor. (4) Arhitectura unor sisteme tolerante la defecte. (5) Abordarea concurentă a defectelor	2	Oral și cu mijloace multimedia, stil de predare interactiv, consultații, implicarea studenților în activități de proiectare.	
05	Modelarea defectelor (1) Definitii: fault, error, failure (2) Evitarea defectelor VS tolerarea defectelor (3) Modele de blocare logica (4) Detectabilitate si redundanta (5) Echivalenta si localizarea defectelor (6) Dominanta. (7) Modelarea erorilor. (8) Modelarea cu modele Markov și lanțuri Markov discrete	2		
06	Generarea vectorilor stimula-test pentru scheme combinate : (1) Metoda activării unei cai (2) Metoda Roth (3) Metoda Poage (4) Elaborarea experimentelor de testare pentru scheme secvențiale (5) Design for testability	2		
07	Analiza indicatorilor de fiabilitate, disponibilitate și mentenabilitate (1) Definitii (2) Fiabilitatea sistemelor serie, paralel și mixte (3) Sisteme duplex tolerante la defecte (4) Disponibilitatea sistemelor de calcul (5) Mentenabilitatea sistemelor de calcul (6) Aplicatii	2		
08	Toleranta la defectare (1) Tehnici de obtinere a redundantei software si hardware (2) Designul sistemelor tolerante la defecte (3) Injectia de erori (4) Sisteme tolerante la defecte. (5) Rețele și sisteme distribuite tolerante la defecte; (a) Heartbeats (b) Timere Watchdog.	2		
09	Metode de proiectare pentru combaterea defectelor. Tehnici de proiectare pentru realizarea toleranței la defectare (1) Conceptul de redundanta (2) Redundanta hardware Structuri redundante pentru implementarea toleranței la defecte în sistemele hardware (1) Viteza de apariție a defectelor hardware (2) Consideratii asupra utilizării redundanței protective în sisteme. (3) Scheme de implementare a redundantei la nivel hardware: (a) Structuri redundante protective statice de tip individual și global rezultate prin multiplicare. (b) Structura redundanță logică majoritară (c) Structura redundanță protectiva statică cu logică cuadruplă / (d) cu logică prin cablare. (e) Structura redundanță prin codare (f) Structura redundanță protectivă dinamică (g) Structura redundanta hibridă. (4) Structuri redundante pentru interconexiunile unui sistem. (5) Probleme de sincronizare în sistemele digitale cu structură redundanță. (6) Criterii de comparare pentru structurile redundante. (a) Indici de îmbunătățire a fiabilității sistemelor cu structură redundanță. (b) Indici de cost și eficiența. (c) Studii de caz privind evaluarea performanțelor unor sisteme cu structură redundanță.	2		
10	Sisteme tolerante la defecte cu redundanță modulară (1) Sisteme M-din-N redundante; (a) Evaluarea fiabilității (b) Circuite de votare. (2) Variații ale redundanței N-modulare; (a) Redundanța modular la nivelul unităților (b) Redundanța dinamică. (3) Redundanța hibridă. (4) Redundanța modular cu filtrare	2		
11	Testarea sistemelor software. Sisteme software embedded. Sisteme digitale total autotestabile. Reconfigurarea sistemelor la apariția defectelor. Model de reconfigurare în sisteme multiprocesor (1) Teste de acceptare. (2) Încapsulările. (3) Reținerea software-ului (4) Toleranță la defecte software bazată pe algoritmi	2		
12	Defecte specifice tehnicii de calcul (a) Defecte de blocare (stuck-at faults) (b) Modelul de scurt-circuit (c) Gate Oxide Shorts (d) Defecte temporare (e)	2		

	Defecte tranzitorii (f) Defecte permanente Perturbatii în sistemele de calcul (a) Consideratii generale (b) Perturbatii datorate diafoniei (c) Perturbatii datorate reflexiilor			
13	Managementul testării: planificarea activităților de testare, dezvoltarea mediului de testare. Evaluarea testării. Redundanța temporală (Checkpoint)	2		
14	Studii de caz . (1) Sistemele NonStop de la TandemComputers. Arhitectura, software, mentenanta (2). Sisteme Stratus (3) Sisteme IBM: G5 si Sysplex (4) Intel Itanium (5) Subsistemele Cassini.	2		
Bibliografie				
1. Cătuneanu V., Bacivarof A. Structuri electronice de inalta fiabilitate. Toleranta la defectări, Ed.Militara Bucuresti 2014 2. Geber T.,s.a. (2018) Fiabilitatea si mentenabilitatea sistemelor de calcul. Ed.Tehnică București 3. Mihalache A. (2017) Cand calculatoarele greșesc... Fiabilitatea sistemelor de programe(software). Ed.didactică si pedagogica București 4. Burlacu E., s.a. (2015) Introducere in studiul fiabilității sistemelor Ed.Științifică București				
Bibliografie minimală				
1) Ștefănescu, C. Sisteme tolerante la defecte, Matrix Rom, Bucuresti 2017 2) Băjenescu, T.I. Fiabilitatea, disponibilitatea si mentenabilitatea sistemelor electronice complexe, Editura de Vest 2017 3) Mihalache, A. Când calculatoarele gresesc. Fiabilitatea sistemelor de programe (software), Ed.Didactică si pedagogică, Bucuresti 2015 4) Popovici, Al.A. Proiectarea securității sistemelor complexe, Ed.Stiintifică si enciclopedică, Bucuresti 1988 5) Israel Koren, C. Mani Krishna, <i>Fault Tolerant Systems</i> , ediția 2, Morgan Kaufmann, eISBN: 9780128181065, 21.09.2020. 6) Cursuri și laboratoare: pe Microsoft Teams				

Lucrările de laborator urmăresc familiarizarea cu principalele tehnici si metode de obtinere a caracteristicii de toleranta la defecte cu ajutorul modelării si simulării. Stadiul lucrărilor de laborator este apreciat la sfirsitul lucrării corespunzatoare, nota la laborator reprezentind media tuturor lucrarilor.

	Aplicații (Seminar / laborator / proiect)	Nr. ore	Metode de predare	Observații
01	Implementarea unui exemplu de redundanță prin replicare fizică a unui acord între elementele replicate folosind algoritmul lui Lamport.	2	Exemple / implementare Prezentare pe tablă si cu mijloace multimedia Experimente si implementare folosind unelte specifice (MS Visual Studio, Diblook)	
02	Implementarea unui exemplu de redundanță informațională utilizând coduri de paritate, sume de verificare, coduri ciclice sau coduri aritmetice.	2		
03	Implementarea unui exemplu de calcul a fiabilității pentru un sistem hibrid serie-paralel.	2		
04	Testare functionala. Unitati de testare automata a sistemelor software bazate pe JAVA cu Junit. Implementarea unui exemplu de toleranță la defecte software.	2		
05	Implementarea unui exemplu de calcul a fiabilității utilizând lanțuri Markov.	2		
06	Determinarea disponibilității sistemelor tolerante la defectări. Determinarea funcției de fiabilitate a sistemelor cu redundanța modulară și a sistemelor autotestabile	2		
07	Implementarea unui exemplu de redundanță temporală. Rețele și sisteme distribuite tolerante la defecte.	2		Evaluarea etapelor de proiectare si implementare
Bibliografie				
1) M. V. Catuneanu, A. Bacivarof - Structuri electronice de intalta fiabilitate. toleranta la defectari - E. M. 2014 2) C. Stefanescu - Sisteme tolerante la defecte - Lit. UPB, 2016 3) Israel Koren, C. Mani Krishna, <i>Fault Tolerant Systems</i> , ediția 2, Morgan Kaufmann, eISBN: 9780128181065, 21.09.2020. 4) Marin L. Shooman: <i>Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design</i> , John Wiley and Sons, 2016. 5) Parag K. Lala. Self-checking and fault-tolerant digital design. Morgan-Kaufmann, 2015.				
Bibliografie minimală				
1. C. Stefanescu - Sisteme tolerante la defecte - Lit. UPB, 2016 2. Israel Koren, C. Mani Krishna, <i>Fault Tolerant Systems</i> , ediția 2, Morgan Kaufmann, eISBN:				

9780128181065, 21.09.2020.
3. Cursuri și laboratoare: pe Microsoft Teams

Observații.

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

<ul style="list-style-type: none"> Se asigură competențe conform prevederilor RNCIS. Studii de caz, exerciții, analiza erorilor
--

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
Curs	Cunoștințe teoretice însușite (cantitatea, corectitudinea, acuratețea) Testarea cunostintelor toretice si a abilități de rezolvare a problemelor	dezbatere Examen scris, Quiz Teste pe parcurs: Evaluare finală: Examen (minim 5)	40%
Seminar/laborator/proiect	Prelucrarea și interpretarea unor rezultate Abilități practice de rezolvare si implementare a problemelor si de proiectare, aplicatii specifice. Prezenta si activitate	Caiet de laborator, Colocviu, Proiect, Portofoliu, exerciții, studii caz. Activitatea de laborator (AL) temele de casă si proiecte (PL); Probe orale, testul docimologic (T)	40%AL+ 10%PL+ 10%T

10.6. Standard minim de performanță

Realizarea efectiva a unei aplicații de tolerare a defectelor pentru o componenta hardware. Modelarea si implementarea unei probleme tipice ingineresti folosind aparatul formal caracteristic domeniului; Studentul trebuie să facă dovada că noțiunile prezentate nu sunt însușite mecanic.

COMPATIBILITATE INTERNATIONALA

- Colorado State University. CS530 – Fault-Tolerant Computing, CS635 – Adv. Fault-Tolerant Computing
- University of Victoria. CSC 454 – Fault Tolerant Computing
- University of Wisconsin-Madison. ECE753 – Fault-Tolerant Computing

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar
	Conf. dr. ing. Cornel POPESCU	Conf. dr. ing. Cornel POPESCU

Data avizării în departament	Semnătura directorului de departament
	Prof. Dr. Ing. Culea George

Data aprobării în Consiliul Facultății	Semnătura decanului
	Conf. dr. ing. Mirela PANAINTE-LEHĂDUȘ