

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	Automatică
1.4 Domeniul de studii	Ingineria sistemelor
1.5 Ciclu de studii	Master
1.6 Programul de studii / Calificarea	Informatică Aplicată
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare Avansată			Codul disciplinei	12.00
2.2 Titularul de curs	Prof.dr.ing. Honoriu VĂLEAN - Honoriu.Valean@aut.utcluj.ro				
2.3 Titularul activităților de seminar / laborator / proiect	Conf.dr.ing. Dan GOȚA – Dan.Gota@aut.utcluj.ro				
2.4 Anul de studiu	2	2.5 Semestrul	1	2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categorica formativă				DA
	Opționalitate				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	3.2 Curs	2	3.3 Seminar		3.3 Laborator	1	3.3 Proiect	1
3.4 Număr de ore pe semestru	56	din care:	3.5 Curs	28	3.6 Seminar	14	3.6 Laborator		3.6 Proiect	14
3.7 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										34
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										10
(d) Tutoriat										3
(e) Examinări										2
(f) Alte activități:										
3.8 Total ore studiu individual (suma (3.7(a)...)3.7(f)))				69						
3.9 Total ore pe semestru (3.4+3.8)				125						
3.10 Numărul de credite				5						

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Complemente de programare, Programarea interfețelor utilizator
4.2 de competențe	N/A

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	N/A
5.2. de desfășurare a seminarului / laboratorului / proiectului	Prezența la laborator și proiect obligatorie

6. Competențele specifice acumulate

Competențe profesionale	• Dezvoltă strategii de soluționare a problemelor • Analizează datele testelor • Asigură conformitatea cu specificațiile • Prezintă rapoarte • Efectuează cercetare științifică • Găsește soluții pentru probleme • Respectă specificațiile contractului • Dezvoltă software cu sursă deschisă • Sintetizează informații • Interacționează profesional în mediile de cercetare și profesionale • Cooperează cu colegii
Competențe transversale	• Dă dovadă de inițiativă • Își asumă responsabilitatea • Gestionează evoluția personală

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul demonstrează cunoașterea: • Conceptelor și principiilor avansate din informatica aplicată, programare, tehnologii digitale și aplicații informatice • Teoriilor și metodologiilor moderne de proiectare, implemetare, testare și mentenanță a aplicațiilor software și a bazelor de date • Modelelor și tehnicilor de analiză a componentelor software • Criteriilor formale de evaluare, comparare și optimizare a aplicațiilor informatice complexe • Metodologiilor moderne de management de proiect • Tehnicilor de analiză a riscurilor, fezabilității și realizabilității proiectelor
Abilități	Studentul va fi capabil să • Analizeze, modeleze și interpreteze aplicații informatice și componentele acestora • proiecteze, implementeze și testeze aplicații software și baze de date utilizând metodologii moderne • identifice și aplice soluții inovative pentru optimizarea aplicațiilor informatice complexe • integreze tehnologii digitale și modele software în ingineria industrială • planifice și gestioneze proiecte software și tehnice • analizeze situații noi folosind cunoștințe multidisciplinare și să anticipeze riscuri
Responsabilitate și autonomie:	Studentul: • Va gestiona autonom proiecte de inginerie software, industrială și sisteme incorporate • Va fi capabil să își asume responsabilitatea pentru calitatea soluțiilor implementate • Va gestiona proiecte profesionale sau de cercetare • Va fi capabil să lucreze eficient în echipe • Va acționa cu inițiativă, adaptabilitate și autonomie în contexte profesionale

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

8.1 Obiectivul general al disciplinei	Deprinderea analizei, proiectării și implementării aplicațiilor software complexe
8.2 Obiectivele specifice	Utilizarea tehnologiilor .net pentru implementarea aplicațiilor complexe

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
Implementarea unei mașini virtuale. Definirea arhitecturii hardware	2 ore	laptop, proiector, dezbateri, curs interactiv / în caz de	
Implementarea unei mașini virtuale. Definirea arhitecturii software	2 ore		

Proiectarea și implementarea unui asamblor simplu. Dictionare	2 ore	forță majoră, on-line platforma Teams	
Proiectarea și implementarea unui asamblor simplu. Regex	2 ore		
Proiectarea și implementarea mașinii virtuale.	4 ore		
Extinderea dicționarului cu instrucțiuni de comparație	2 ore		
Reimplementarea multifilară a mașinii virtuale. Trasarea manual a programelor	2 ore		
Extinderea dicționarului cu instrucțiuni de incrementare / decrementare și de deplasare	2 ore		
Fanioane virtuale. Implementarea instrucțiunilor de salt	2 ore		
Extinderea dicționarului cu instrucțiuni aritmetice și de manipulare a fanioanelor	2 ore		
Implementarea stivei. Instrucțiuni PUSH-POP pe octet și cuvânt	2 ore		
Funcții	2 ore		
Implementarea de programe în limbaj de asamblare.	2 ore		
Implementarea unei mașini virtuale. Definirea arhitecturii hardware	2 ore		
Bibliografie 1. H. Valean. Programare avansata. https://users.utcluj.ro/~valean/courses/VM/MV.pdf 2. Sz. Enyedi, I. Lengyel, L. Miclea, I. Stefan, O. Stan, H. Valean. Dezvoltarea si testarea aplicatiilor software. Ed. Risoprint, 2014. 3. A. Troelsen, P. Japikse. C# 6.0 and the .NET 4.6 Framework. APRESS, 2015. 4. W. De Kort. Exam Ref 70-483: Programming in C#. O'Reilly Media, 2013. 5. How to create your own virtual machine. https://www.codeproject.com/Articles/43176/How-to-create-your-own-virtual-machine 6. How to create your own virtual machine Part 2. https://www.codeproject.com/Articles/61924/How-to-create-your-own-virtual-machine-Part-2			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Limbaje de asamblare. Exemple	1 oră	Prezentare de exemple, discuții, aplicații practice / în caz de forță majoră, on-line platforma Teams	Prezența obligatorie
Dicționare. Aplicații bazate pe dicționare	1 oră		
Regex. Aplicații ale Regex	1 oră		
Programare multifilara	1 oră		
Trasabilitatea manuala a programelor	1 oră		
Dezvoltarea de programe in limbaj de asamblare și executarea lor pe mașina virtuală	2 ore		
Limbaje de asamblare. Exemple	1 oră		
Bibliografie 1. H. Valean. Programare avansata. https://users.utcluj.ro/~valean/courses/VM/MV.pdf 2. Sz. Enyedi, I. Lengyel, L. Miclea, I. Stefan, O. Stan, H. Valean. Dezvoltarea si testarea aplicatiilor software. Ed. Risoprint, 2014. 3. A. Troelsen, P. Japikse. C# 6.0 and the .NET 4.6 Framework. APRESS, 2015. 4. W. De Kort. Exam Ref 70-483: Programming in C#. O'Reilly Media, 2013. 5. How to create your own virtual machine. https://www.codeproject.com/Articles/43176/How-to-create-your-own-virtual-machine 6. How to create your own virtual machine Part 2. https://www.codeproject.com/Articles/61924/How-to-create-your-own-virtual-machine-Part-2			

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

Conținutul disciplinei, împreună cu deprinderile și abilitățile dobândite, corespund așteptărilor organizațiilor profesionale de profil, firmelor de profil, precum și a organismelor naționale și internaționale de asigurare a calității (ARACIS). De asemenea asigură adoptarea unor standarde etice adecvate practicii ingineresti.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Evaluarea cunoștințelor pe baza prezentării unei aplicații și discuții asupra acesteia	Examen oral (examinare sumativă)	60%
11.5 Seminar/Laborator /Proiect	Examinarea deprinderilor și cunoștințelor practice obținute în urma participării la laborator. Prezentarea și susținerea proiectului realizat	Verificare lucrări (evaluare continuă) / prezentare proiect (evaluare sumativă)	20% / 20%
11.6 Standard minim de performanță Standard minim de performanță: Notă examen > 5 și notă colocviu laborator > 5 și notă proiect > 5			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
01.07.2026	Curs	Prof.dr.ing. Honoriu VĂLEAN	
	Aplicații	Conf.dr.ing. Dan GOȚA	

Data avizării în Consiliul Departamentului de Automatică	Director Departament Automatică Prof.dr.ing. Honoriu VĂLEAN

Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan Prof.dr.ing. Vlad MUREȘAN
