

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Facultatea de Automatică și Calculatoare
1.3 Departamentul	Automatică
1.4 Domeniul de studii	Ingineria Sistemelor
1.5 Ciclul 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	Complemente de programare				Codul disciplinei	2.00
2.2 Titularul de curs	Prof. dr. ing. Liviu Miclea – Liviu.Miclea@aut.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	S.I.dr. Cosmina Emilia Corches – Cosmina.Corches@aut.utcluj.ro					
2.4 Anul de studiu	1	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	3	din care:	3.2 Curs	2	3.3 Seminar	0	3.3 Laborator	1	3.3 Proiect	0
3.4 Număr de ore pe semestru	42	din care:	3.5 Curs	28	3.6 Seminar	0	3.6 Laborator	14	3.6 Proiect	0
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										23
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										22
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										31
(d) Tutoriat										4
(e) Examinări										3
(f) Alte activități:										0
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))				83						
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	N/A
4.2 de competențe	Cunoștințe de utilizarea calculatoarelor

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	
5.2. de desfășurare a seminarului / laboratorului / proiectului	

6. Competențele specifice acumulate

Competențe profesionale	• Dezvoltă strategii de soluționare a problemelor; • Asigură conformitatea cu specificațiile; • Găsește soluții pentru probleme; • 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, implementare, testare și mentenanță a aplicațiilor software și a bazelor de date - Modelelor și tehnicilor de analiză a componentelor software
Abilități	Studentul va fi capabil să: - Analizeze, modeleze și interpreteze aplicații informatice și componentele acestora - Sintetizeze și interpreteze date și rezultate ale testelor - Compare alternative de proiectare, optimizare și implementare pe baza unor criterii formale - proiecteze, implementeze și testeze aplicații software și baze de date utilizând metodologii moderne - planifice și gestioneze proiecte software și tehnice - Elaboreze rapoarte, prezentări și documentații tehnice profesionale - Comunice și colaboreze eficient în echipe multidisciplinare
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 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	Obiectivul principal al acestei disciplinei transmiterea de cunoștințe de bazele programării calculatoarelor, familiarizarea cu diferite medii și limbaje de programare și instruirea în vederea realizării de aplicații software complexe.
8.2 Obiectivele specifice	Studentii vor învăța să: - Utilizeze cunoștințe de programarea calculatoarelor - Realizeze proiecte software avansate pentru implementarea aplicațiilor informatice

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
1. Noțiuni privind reprezentarea în calculator a informației	2	Expunerea didactică, conversația didactică, problematizarea	
2. Programarea aplicațiilor simple în C și C++	2		
3. Dezvoltarea în Visual C++ de aplicații tip Console	2		
4. Tipuri de date	2		
5. Instrucțiunile limbajelor de programare	2		
6. Programarea aplicațiilor complexe, pointeri, tablouri	2		
7. Structura programului și declararea variabilelor. Referințe.	2		
8. Date structurate. Operații cu fișiere.	2		
9. Funcții din biblioteca limbajului C	2		
10. Algoritmi fundamentali	2		
11. Programare obiectuală în Java	2		
12. Clase, obiecte, clase derivate	2		
13. Realizarea aplicațiilor complexe în Java. Reutilizarea claselor	2		
14. Realizarea unei interfete	2		

Bibliografie 1. Negrescu L. Limbajele C si C++ pentru incepatori. Cota 504.360/2. 2001 2. Breazu M., Programare orientata pe obiecte : principii : C++. Cota 513.643. 2002 3. Williams M, Bennet D. Visual C++6 Unleashed. Cota 502.554, 2002. 4. Andrews M. Visual C++ object-oriented programming. Cota 479.013, 1993 5. Striletschi C. et all. Tehnologii obiectuale si algoritmi de baza privind dezvoltarea aplicatiilor in limbajul de programare C/C++. Cota 520.986, 2007. 6. Pop P. et all. Tehnologii practice privind dezvoltarea aplicatiilor in medii de programare C/C++. Cota 520.495, 2006 7. L. Miclea, M. Damian, Complemente de programare, note de curs (format electronic)			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
L1,2. Prezentarea mediului de programare Visual Studio	2	Modelarea, învățarea prin descoperire, exercițiul	
L3,4. Transformarea unor probleme în algoritmi și implementarea aplicațiilor folosind tipurile de bază și instrucțiunile limbajului.	2		
L5,6. Implementarea unor aplicații cu funcții	2		
L7,8. Implementarea unei aplicații care conține o listă simplu-înlanțuită	2		
L9,10. Implementarea unei aplicații care utilizează fișiere	2		
L11,12. Implementarea unor aplicații cu clase și obiecte.	2		
L13,14. Implementarea unor aplicații care utilizează ierarhii de clase.	2		
Bibliografie 1. Negrescu L. Limbajele C si C++ pentru incepatori. Cota 504.360/2. 2001 2. Breazu M., Programare orientata pe obiecte : principii : C++. Cota 513.643. 2002 3. Williams M, Bennet D. Visual C++6 Unleashed. Cota 502.554, 2002. 4. Andrews M. Visual C++ object-oriented programming. Cota 479.013, 1993 5. Striletschi C. et all. Tehnologii obiectuale si algoritmi de baza privind dezvoltarea aplicatiilor in limbajul de programare C/C++. Cota 520.986, 2007. 6. Pop P. et all. Tehnologii practice privind dezvoltarea aplicatiilor in medii de programare C/C++. Cota 520.495, 2006 7. L. Miclea, M. Damian, Complemente de programare, note de curs (format electronic)			

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

Disciplina este fundamentală pentru o carieră în informatică și programarea calculatoarelor. Conținutul îmbină cunoștințe teoretice cu aplicații și se concentrează pe formularea și rezolvarea unor probleme specifice care pot apare într-o diversitate de domenii din inginerie.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Abilitatea rezolvării problemelor	Examen oral: teoretic (20%) și practic (80%)	100%
11.5 Seminar/Laborator /Proiect	Abilitatea rezolvării problemelor. Prezență. Interactivitate.	Verificari pe parcurs	A/R
11.6 Standard minim de performanță Rezolvarea problemelor de programare utilizând limbajele de programare specifice disciplinei. Laborator: Admis, Examen oral: N(teorie)≥5 și N(practic)≥5.			

Data completării: 10.07.2026	Titulari	Titlu Prenume NUME	Semnătura
	Curs	Prof.dr.ing. Liviu Miclea	
	Aplicații	S.I.dr. Cosmina Emilia Corcheș	

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