

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	Licență
1.6 Programul de studii / Calificarea	Calculatoare și Tehnologia Informației / Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Tehnici de Programare Fundamentale				Codul disciplinei	25
2.2 Titularul de curs	Prof. dr. ing. Cioara Tudor - Tudor.Cioara@cs.utcluj.ro Conf. dr. ing. Chifu Viorica - Viorica.Chifu@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	As. dr. ing. Mitrea Dan - Dan.Mitrea@cs.utcluj.ro As. ing. Rancea Alexandru - Alexandru.Rancea@cs.utcluj.ro As. ing. Antonesi Gabriel - Gabriel.Antonesi@cs.utcluj.ro					
2.4 Anul de studiu	II	2.5 Semestrul	4	2.6 Tipul de evaluare	E	
2.7 Regimul disciplinei	Categorii formative: <i>DF – fundamentală, DS – de specialitate, DC – complementară</i>					DF
	Opționalitate: <i>DOB – obligatorie, DOP – opțională, DFA – facultativă</i>					DOB

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	Curs	2	Seminar	-	Laborator	2	Proiect	-
3.2 Număr de ore pe semestru	56	din care:	Curs	28	Seminar	-	Laborator	28	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										10
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminar / laboratoare, teme, referate, portofolii și eseuri										20
(d) Tutoriat										-
(e) Examinări										4
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))				44						
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	Bazele Programării Orientate pe Obiect, Structuri de Date și Algoritmi
4.2 de competențe	Cunoștințe legate de programare orientată pe obiect

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator;
5.2. de desfășurare a laboratorului	Tabla, proiector, calculator, internet, software specific;

6. Competențele specifice acumulate

6.1 Competențe profesionale	C4 - Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații • C4.1 - Identificarea și descrierea elementelor definitorii ale performanțelor sistemelor hardware, software și de comunicații • C4.2 - Explicarea interacțiunii factorilor care determină performanțele sistemelor hardware, software și de comunicații • C4.3 - Aplicarea metodelor și principiilor de bază pentru creșterea performanțelor sistemelor hardware, software și de comunicații • C4.4 - Alegerea criteriilor și metodelor de evaluare a performanțelor sistemelor hardware, software și de comunicații • C4.5 - Dezvoltarea de soluții profesionale pentru sisteme hardware, software și de comunicații bazate pe creșterea performanțelor
6.2 Competențe transversale	N/A

7. Rezultatele așteptate ale învățării

Cunoștințe	Describe, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare, baze de date, și inginerie software și modul lor de aplicare în probleme concrete.
Aptitudini	• Alege și explică concepte proprii specifice proiectării algoritmilor, programării orientate pe obiecte, și programării funcționale. • Specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbaje de programare obiect-orientate (Java), aplicând elementele specifice ingineriei software. • Proiectează, implementează, dezvoltă aplicații ce includ baze de date.
Responsabilitate și autonomie:	Are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	Cunoașterea și aplicarea tehnicilor de programare orientate pe obiect în dezvoltarea aplicațiilor software
8.2 Obiectivele specifice	- Sa poata utiliza tehnici de programare pentru proiectarea claselor si interfețelor, incluzand contracte și invarianti. - Sa poata utiliza tehnici de programare pentru reutilizarea codului folosind mostenirea si polimorfismul. - Sa poata utiliza tehnici de programare bazate pe generics si stream-uri pentru procesarea colectiilor. - Sa poata utiliza tehnici de programare pentru reflection, design pattern-uri si framework-uri in vederea reutilizarii solutiilor de proiectare. - Sa aplice principiile de proiectare SOLID si thread-urile in Java. - Sa poata utiliza tehnici de programare orientate-obiect si functionale intr-o abordare integrata pentru dezvoltarea de programe flexibile si eficiente. - Sa poata utiliza expresiile lambda si sa poata efectua operatii de procesare a stream-urilor.

9. Conținuturi

9.1 Curs	Nr.ore	Metode de predare	Observații
Procesul de dezvoltare software si paradigme de programare	2	- Folosirea metodelor multimedia de predare si	N/A
Modelare UML, Pattern-uri arhitecturale	2		

Flexibilitate si reutilizare folosind Design Patterns	2	acces la Internet - Prezentarea cursurilor si discutii. - Intrebări adresate studenților în timpul cursului.	
Paradigme de programare orientate pe obiect	4		
Principii SOLID, inversarea controlului și framework-uri	2		
Practici recomandate în dezvoltarea software orientată pe obiecte	2		
Tehnici de programare cu fire de execuție	4		
Tehnici de programare bazate pe tipuri generice și reflecție	4		
Tehnici de programare cu expresii lambda și procesare de fluxuri de date	4		
Tehnici de programare folosind contracte și invarianți	2		

Bibliografie:

1. B. Eckel, On Java 8, MindView LLC, 2017
2. E. Gamma, R. Helm, R. Johnson, J. Vlissides - Design Patterns, Addison Wesley Professional, 1994
3. K. Sharan, P. Späth, More Java 17: An In-Depth Exploration of the Java Language and Its Features 3rd Edition, Apress, 2021
4. R. Urma, M. Fusco, A. Mycroft, Modern Java in Action: Lambdas, streams, functional and reactive programming, 2nd Edition, Manning, 2018
5. Note de curs online puse la dispozitie de titularii de curs
- Online : <http://docs.oracle.com/javase/tutorial/index.html>
<http://stackoverflow.com/>

9.2 Aplicații - seminar / laborator / proiect	Nr.ore	Metode de predare	Observații
Introducere în resursele și cerințele laboratorului	2	Scurtă prezentare a temelor de laborator, discuții privind cerințele acestora și implementarea temelor pe calculator.	N/A
Tema 1 – Tehnici de programare bazate pe moștenire și polimorfism	4		
Tema 2 – Tehnici de programare cu fire de execuție	6		
Tema 3 – Tehnici de programare bazate pe reflecție, tipuri generice și procesarea fluxurilor de date	6		
Evaluarea temelor	8		
Colocviu de laborator	2		

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*)

- Online : <http://docs.oracle.com/javase/tutorial/index.html>
<http://stackoverflow.com/>

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

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

Tehnici Fundamentale de Programare este o disciplină din domeniul „Calculatoare și Tehnologia Informației”. Aceasta îi învață pe studenți să aplice tehnici de programare orientată pe obiecte în proiectarea și implementarea aplicațiilor software. Conținutul disciplinei a fost elaborat pe baza analizei unor discipline similare din alte universități, precum și pe baza cerințelor angajatorilor din domeniul IT. De asemenea, conținutul a fost evaluat de agențiile guvernamentale românești CNEAA și ARACIS.

11. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Cunoașterea și utilizarea tehnicilor de programare prezentate în cadrul cursurilor; prezența și interacțiunea la cursuri.	Examen scris.	50%
Seminar	-	-	-

Laborator	- Abilitatea de proiectare si implementare a programelor orientate pe obiect. - Abilitatea de a aplica tehnici de programare in practica. - Calitatea codului scris si a diagramelor UML realizate in etapa de proiectare. - Activitatea si prezenta in cadrul sesiunilor de laborator.	Evaluarea temelor de laborator pe parcursul semestrului fata in fata. Colocviu de laborator.	50%
Proiect	-	-	-

Standard minim de performanță: Sa poata utiliza tehnici de programare orientate pe obiect in proiectarea si implementarea aplicatiilor software.

Conditii de promovare a laboratorului:

- Predarea la timp a tuturor temelor de laborator.
- Minim nota 5 pe fiecare tema.
- Prezenta la cel putin 11 lucrari de laborator.
- Medie note teme laborator ≥ 5 ; Nota colocviu ≥ 5

Calcul nota laborator: 50% media notelor pe teme + 50% nota colocviu.

Conditii de participare la examenul final: promovarea laboratorului.

Calcul nota disciplina: 50% laborator + 50% examen final.

Conditii de promovare disciplina: Examen final ≥ 5

Conditii de predare teme de laborator restante: o singura tema poate fi amanata si prezentata intr-una dintre sesiunile de restante.

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
30.04.2026	Curs	Prof.dr.ing. Tudor CIOARA	
		Conf.dr.ing. Viorica Chifu	
	Aplicații	As. dr. ing. Dan MITREA	
		As. drd. ing. Alexandru RANCEA	
		As. drd. ing. Gabriel Antonesi	

Data avizării în Consiliul Departamentului Calculatoare

Director Departament,
Prof.dr.ing. Rodica Potolea

Data aprobării în Consiliul Facultății de Automatică și Calculatoare

Decan,
Prof.dr.ing. Vlad Mureșan