

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 Ciclul 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	Programarea Calculatoarelor				Codul disciplinei	5
2.2 Titularul de curs	Conf. dr. ing. Giosan Ion - Ion.Giosan@cs.utcluj.ro Șl. dr. Costin Iulia - Iulia.Costin@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	Conf. dr. ing. Ion Giosan - Ion.Giosan@cs.utcluj.ro Șl. dr. Costin Iulia - Iulia.Costin@cs.utcluj.ro					
2.4 Anul de studiu	I	2.5 Semestrul	1	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	5	din care:	Curs	2	Seminar	1	Laborator	2	Proiect	-
3.2 Număr de ore pe semestru	70	din care:	Curs	28	Seminar	14	Laborator	28	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										30
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										25
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										13
(d) Tutoriat										7
(e) Examinări										5
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f))				80						
3.5 Total ore pe semestru (3.2+3.4)				150						
3.6 Numărul de credite				6						

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	N/A
4.2 de competențe	N/A

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Amfiteatru mare, materiale suport: tabla, calculator, videoproiector
5.2. de desfășurare a seminarului / laboratorului	Laborator cu calculatoare, tabla. Software specific (MinGW C, Code::Blocks)

6. Competențele specifice acumulate

6.1 Competențe profesionale	C1 - Operarea cu fundamente matematice, ingineresti și ale informaticii C1.1 - Recunoașterea și descrierea conceptelor proprii calculabilității, complexității, paradigmelor de programare și modelării sistemelor de calcul și comunicații C1.2 - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea structurii și funcționării sistemelor hardware, software și de comunicații C1.3 - Construirea unor modele pentru diferite componente ale sistemelor de Calcul C1.4 - Evaluarea formală a caracteristicilor funcționale și nefuncționale ale sistemelor de calcul C1.5 - Fundamentarea teoretică a caracteristicilor sistemelor proiectate
6.2 Competențe transversale	N/A

7. Rezultatele așteptate ale învățării

Cunoștințe	Describe, identifică și sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, și anume: - Conceptele de bază ale programării procedurale: execuția secvențială a instrucțiunilor, utilizarea structurilor de control, organizarea codului în funcții, folosirea variabilelor și tipurilor de date, precum și modularitatea programelor - Conceptele de bază ale limbajului de programare C: tipuri de date, variable, expresii, instrucțiuni, funcții, programare modulară, pointeri, managementul memoriei, șiruri de caractere, tipuri de date structură, uniune, enumerare, tipuri de date utilizator, lucrul cu fișiere, recursivitate, argumente la execuția programelor. - Mediul de dezvoltare Code::Blocks pentru punerea în practică a conceptelor programării procedurale în diverse probleme aplicative, utilizând limbajul de programare C.
Aptitudini	Alege și explică concepte proprii specifice programării în limbajul de programare C. Specifică, elaborează algoritmi, dezvoltă și testează programe în limbajul de programare C. Utilizează mediul de programare Code::Blocks si unelte de depanare specifice programării in limbajul de programare C.
Responsabilitate și autonomie:	Demonstrează autonomie în aplicarea cunoștințelor și metodelor specifice domeniilor Calculatoare si Tehnologia Informației, își asumă responsabilitatea pentru calitatea soluțiilor tehnice propuse și pentru respectarea standardelor profesionale, legale și etice.

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	Învățarea utilizării unui limbaj de nivel înalt pentru scrierea de programe
8.2 Obiectivele specifice	- Înțelegerea problemelor de dimensiuni reduse expuse în limbaj natural și dezvoltarea unor soluții sub forma programelor de calculator; - Înțelegerea codului sursă scris de alți programatori și abilitatea de a analiza critic acel cod; - Proiectarea și implementarea programelor în limbajul C folosind o abordare structurată / modulară; - Învățarea unui stil de programare adecvat; - Identificarea cauzelor erorilor de programare și corectarea acestora.

9. Conținuturi

9.1 Curs	Nr.ore	Metode de predare	Observații
Limbaje de programare. Etapele rezolvării problemelor utilizând calculatorul. Definirea și proprietățile algoritmilor. Limbajul C - caracteristici. Tipuri de date. Funcții de intrare / ieșire	2		
Stil de programare. Reprezentarea internă a datelor. Variabile și expresii.	2		

Instrucțiuni în C. Preprocesarea în C	2	Cursuri, demonstrații și discuții.	Utilizarea unui videoproiector
Funcții (structură, apel, transmiterea argumentelor, funcții ca și parametru, domeniul variabilelor). Funcții pentru procesarea caracterelor	2		
Programare modulară. Depanarea programelor	2		
Pointeri. Managementul memoriei	2		
Pointeri și tablouri. Pointeri la funcții	2		
Șiruri de caractere în C. Biblioteca C	2		
Tipurile structură, uniune, enumerare. Definirea tipurilor utilizator	2		
Lucrul cu fișiere. Funcții de intrare/ieșire de nivel înalt	2		
Recursivitate. Mecanism și exemple	2		
Lucrul cu timpul, redirectarea intrării/ieșirii. Liste de argumente cu dimensiune variabilă. Argumente la execuția programelor. Structuri autoreferențiale	2		
Exemple de programe explicate. (Generare combinatorie, algoritmi simpli de sortare, etc.)	2		
Recapitulare	2		
Bibliografie: 1. I. Ignat, C.L. Ignat. Programarea calculatoarelor. Descrierea algoritmilor și fundamentele limbajului C/C++. Ed. Albastră, Cluj-Napoca, 2005, I.S.B.N. 973-650-163-9. 2. Paul and Harvey Deitel, C: How to program, Pearson Education, 6ed, 2010 3. K.N. King, C Programming: A modern Approach, W.W. Norton, 2008 4. Stephen Prata, C Primer Plus, Sams, 5ed, 2004 5. Brian W. Kernighan, Dennis M. Ritchie – The C Programming Language, Prentice Hall, Inc., 1988.			
9.2 Aplicații - seminar / laborator	Nr.ore	Metode de predare	Observații
S1.Reprezentarea algoritmilor (scheme logice și pseudocod)	1	Îndrumare, discuții, rezolvări de probleme. Îndrumare, discuții și dezvoltarea asistată a programelor.	Calculatoare echipate cu MinGW C și mediul de dezvoltare Code::Blocks
S2. Operatori, expresii și instrucțiuni	1		
S3. Funcții și programare modulară	1		
S4. Pointeri și gestiunea memoriei	1		
S5. Șiruri de caractere. Argumente la execuția programelor	1		
S6. Tipurile structură, uniune și enumerare	1		
S7. Recursivitate. Lucrul cu fișiere	1		
L1. Pseudocod. Medii de dezvoltare pentru programarea în C. Instalarea și folosirea mediului Code::Blocks.	2		
L2. Funcții de intrare / ieșire în C	2		
L3. Expresii în C	2		
L4. Instrucțiuni în C	2		
L5. Funcții. Depanarea programelor C	2		
L6. Programare modulară	2		
L7. Pointeri. Pointeri și tablouri	2		
L8. Alocarea memoriei. Pointeri la funcții	2		
L9. Șiruri de caractere	2		
L10. Tipurile structură, uniune, enumerare	2		
L11. Funcții de nivel înalt pentru lucrul cu fișiere în C	2		
L12. Recursivitate	2		
L13. Recapitulare	2		
L14. Test de laborator	2		

Bibliografie:

1. I. Ignat. Programarea calculatoarelor. Îndrumător de lucrări de laborator. Ed. U.T.Pres, Cluj-Napoca, 2003, ISBN 973-662-024-7.
2. Note de curs, lucrări de laborator, materiale pentru seminarii:
http://users.utcluj.ro/~igiosan/teaching_pc.html
<http://users.utcluj.ro/~robert>

**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

Conținutul cursului este în concordanță cu recomandările curriculumului ACM pentru știința calculatoarelor.

11. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Concepte de programare în limbajul C Abilități de rezolvare a problemelor folosind limbajul de programare C	Teste la curs (T) + Examen final scris (E)	60% = 50% E + 10% T
Seminar	Rezolvare de probleme folosind limbajul de programare C	Verificarea temelor	-
Laborator	Calitatea aplicațiilor de laborator și evaluarea testului de laborator	Evaluarea rezolvării problemelor la test	40%
Proiect	-	-	-

Standard minim de performanță:

Calcul nota disciplina: 40% laborator + 60% examen final și teste

Conditii de participare la examenul final: Laborator ≥ 5 , Seminar (teme realizate)

Conditii de promovare: Nota disciplina ≥ 5

Data completării: 27.04.2026	Titulari	Titlu, prenume / nume	Semnătura
	Curs	Conf. dr. ing. Ion GIOSAN	
		Șl. dr. Iulia COSTIN	
	Aplicații	Conf. dr. ing. Ion GIOSAN	
		Șl. dr. Iulia COSTIN	

Data avizării în Consiliul Departamentului

Director Departament,
Prof.dr.ing. Rodica Potolea

Data aprobării în Consiliul Facultății de Automatică și Calculatoare

Decan,
Prof.dr.ing. Vlad Mureșan