FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Inginerie Electrică și Știința Calculatoarelor
Departamentul	Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licență, dual
Programul de studii	Calculatoare

2. Date despre disciplină

Denumirea disciplinei	**PROGRAMAREA CALCULATOARELOR ȘI LIMBAJE DE PROGRAMARE**				
Anul de studiu	**I**	Semestrul	**1**	Tipul de evaluare	**E**
Regimul disciplinei	Categoria formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				**DF**
	Categoria de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				**DOB**

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	**5**	Curs	**3**	Seminar		Laborator/ Lucrări practice IIS	**2**	Proiect IIS		Practică IIS	
						Laborator/ Lucrări practice IM		Proiect IM		Practică IM	
I b) Totalul de ore pe semestru din planul de învățământ	**70**	Curs	**42**	Seminar		Laborator	**28**	Proiect		Practică	

(*IIS – instituție de învățământ superior; IM – învățare prin muncă*)

Distribuția fondului de timp pe semestru	Ore IIS	Ore IM
II.Studiu individual	**27**	**0**
III. Examinări	**3**	**0**
IV. Alte activități (precizați):	**0**	**0**

Total ore studiu individual (II + III)	Ore IIS	**30**	Ore IM	**0**
Total ore pe semestru (Ib+II+III+IV)	Ore IIS	**100**	Ore IM	**0**
Numărul de credite	Credite IIS	**4**	Credite IM	**0**

4. Competențe specifice acumulate

Competențe transversale	CT5. aplică cunoștințe științifice, tehnologice și inginerești
Competențe profesionale	CP03. definește arhitectura software CP04. creează softuri CP05. dezvoltă prototipul pentru software CP12. utilizează biblioteci de software

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/absolventul identifică și descrie concepte, principii și metode de bază din informatică: algoritm, program, tip de dată, structură de control, funcție.	Studentul/absolventul rezolvă probleme cu aplicabilitate în inginerie prin programe în limbajul C și validează soluția obținută prin testare cu date impuse.	Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluarea în luarea deciziilor, alegând și justificând soluția algoritmică adecvată.
Studentul/absolventul explică și interpretează rezultatele execuției programelor C, inclusiv mesajele de eroare la compilare și execuție.	Studentul/absolventul achiziționează și prelucrează date de la tastatură și din fișiere text sau binare și interpretează rezultatele obținute.	Studentul/absolventul comunică eficient despre activitățile de inginerie, prezentând și argumentând soluțiile proprii la susținerea temei de casă.

Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare și tehnici de programare structurată, procedurală și modulară și modul lor de aplicare în probleme concrete.	Studentul/absolventul dezvoltă și testează programe în limbajul C de complexitate medie, utilizând tablouri, șiruri de caractere, funcții, pointeri, structuri și fișiere, organizate modular.	Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei, elaborând cod sursă propriu și utilizând responsabil instrumentele de inteligență artificială.

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

Obiectivul general al disciplinei	Însușirea elementelor fundamentale privind paradigmele de programare structurată, procedurală și modulară și asimilarea fundamentelor limbajului de programare C: tipuri de date, structuri de control, funcții, pointeri și alocare dinamică, tipuri de date agregat și lucrul cu fișiere. O1. Prezentarea conceptelor de algoritm, program și programare structurată; O2. Asimilarea elementelor de bază ale limbajului C: tipuri de date, operatori, expresii, instrucțiuni; O3. Utilizarea tablourilor și a șirurilor de caractere în rezolvarea problemelor; O4. Deprinderea programării procedurale prin definirea și utilizarea funcțiilor în C; O5. Utilizarea pointerilor și a alocării dinamice a memoriei; O6. Structurarea datelor utilizând structuri, uniuni, câmpuri de biți și enumerări; O7. Prelucrarea datelor persistente prin fișiere text și binare; O8. Deprinderea programării modulare prin dezvoltarea, testarea și depanarea unor aplicații proiect.

7. Conținutul predării și învățării

Curs	Nr. ore	Metode de predare	Observații
1. Noțiuni introductive 1.1. Structura și funcționarea unui sistem de calcul 1.2. Algoritmi, programe, limbaje de programare 1.3. Etapele rezolvării unei probleme într-un limbaj de programare: editare, compilare, link-editare, execuție, depanare 1.4. Limbajul C: evoluție, standarde (C99, C11, C17, C23), medii de dezvoltare	3	expunerea, prelegerea-dezbatere, demonstrația	
2. Elemente de bază ale limbajului C 2.1. Structura și sintaxa programelor C. Directivele #include și #define 2.2. Alfabet, identificatori, cuvinte-cheie, comentarii 2.3. Tipuri fundamentale de date. Constante, variabile, declarații 2.4. Operații de intrare/ieșire standard (printf, scanf, getchar, putchar)	3	expunerea, prelegerea-dezbatere, demonstrația	
3. Operatori, expresii și instrucțiuni de control 3.1. Operatori și expresii. Precedența și asociativitatea operatorilor 3.2. Conversii de tip implicite și explicite 3.3. Instrucțiuni care implementează structurile de control: secvența, decizia (if, switch), repetiția (while, do-while, for) 3.4. Instrucțiunile break, continue, return. Programare structurată versus programare cu goto	3	expunerea, prelegerea-dezbatere, demonstrația	
4. Tablouri de date 4.1. Tablouri unidimensionale (vectori) și bidimensionale (matrice) 4.2. Declararea și inițializarea tablourilor 4.3. Citirea și afișarea tablourilor 4.4. Algoritmi elementari cu tablouri: căutare, sortare, operații cu matrice	3	expunerea, prelegerea-dezbatere, demonstrația	
5. Caractere și șiruri de caractere 5.1. Tipul char. Constante și variabile de tip caracter 5.2. Șiruri de caractere: declarare, inițializare, citire și afișare 5.3. Funcții standard pentru caractere (ctype.h) și pentru șiruri de caractere (string.h) 5.4. Tablouri de șiruri de caractere 5.5. Conversii între șiruri de caractere și tipurile numerice	3	expunerea, prelegerea-dezbatere, demonstrația	
6. Funcții în C. Programare procedurală 6.1. Principiile programării procedurale 6.2. Declararea (prototipul), definirea și apelul funcțiilor 6.3. Parametri formali și parametri efectivi. Transmiterea parametrilor prin valoare 6.4. Parametri de tip tablou și șir de caractere 6.5. Domeniul de vizibilitate și durata de viață a variabilelor 6.6. Recursivitate în C 6.7. Funcții cu număr variabil de parametri (stdarg.h)	6	expunerea, prelegerea-dezbatere, demonstrația	
7. Pointeri 7.1. Tipul de dată pointer. Operatorii & și * 7.2. Operații cu pointeri. Aritmetica pointerilor 7.3. Pointeri și tablouri. Pointeri și șiruri de caractere 7.4. Transmiterea parametrilor prin adresă 7.5. Pointeri la funcții 7.6. Alocarea dinamică a memoriei (malloc, calloc, realloc, free). Erori frecvente de gestionare a memoriei	6	expunerea, prelegerea-dezbatere, demonstrația	

8. Tipuri de date agregat în C 8.1. Structuri: declarare, inițializare, acces la membri 8.2. Tablouri de structuri. Pointeri la structuri și alocare dinamică 8.3. Uniuni. Câmpuri de biți 8.4. Tipuri enumerare. Definirea tipurilor cu typedef	3	expunerea, prelegerea-dezbatere, demonstrația	
9. Fișiere și lucrul cu fișiere în limbajul C 9.1. Fluxuri și fișiere. Deschiderea și închiderea fișierelor 9.2. Fișiere text: citire și scriere formatată, pe caractere și pe linii 9.3. Fișiere binare: citire și scriere de blocuri (fread, fwrite) 9.4. Acces direct (fseek, ftell). Tratarea erorilor la lucrul cu fișiere 9.5. Aplicații: prelucrarea fișierelor de înregistrări	6	expunerea, prelegerea-dezbatere, demonstrația	
10. Programare modulară în C 10.1. Clase de memorare (auto, register, static, extern) 10.2. Proiecte cu mai multe fișiere sursă. Fișiere antet și compilare separată 10.3. Transmiterea datelor între module. Directive de compilare condiționată 10.4. Biblioteci de subprograme. Construirea proiectelor (make) 10.5. Parametri în linia de comandă 10.6. Dezvoltarea unei aplicații modulare complete	6	expunerea, prelegerea-dezbatere, demonstrația	
Bibliografie minimală recomandată			
1. Jeff Szuhay, Learn C Programming (2nd Edition), Packt Publishing, ISBN-13: 978-1801078450, 2022; 2. Stephen G. Kochan, Programming in C (4th Edition) (Developer's Library), Addison-Wesley Professional, ISBN-13: 978-0321776419, 2014; 3. L. Negrescu - Limbajele C și C++ pentru începători, vol. I, Editura Albastră, Cluj-Napoca, 2012; 4. Brian Kernighan, Dennis Ritchie, The C Programming Language (2nd Edition), Prentice Hall, 1988; ISBN-13: 978-0131103627; 5. K. Jamsa, L. Klander - Totul despre C și C++, Editura Teora, 2013; 6. Remus Prodan – pagina web a cursului: http://dev.fiesc.usv.ro/remus; 7. B.W. Kernighan, D.M. Ritchie - Programarea în limbaj C, Editura Teora, 2003; 8. K. N. King, C Programming: A Modern Approach (2nd Edition), W. W. Norton & Company, ISBN-13: 978-0393979503, 2008; 9. Pentiuc St. Gh., Turcu Cristina, Turcu Cornel, Mahalu George, Petrișor Cristinel - Programarea calculatoarelor, Universitatea „Ștefan cel Mare” Suceava, 1995.			

Aplicații (seminar / laborator / lucrări practice / proiect)	Nr. ore	Metode de predare	Observații
1. Prezentarea generală a laboratorului în care se vor desfășura activitățile practice pentru disciplina Programarea calculatoarelor și limbaje de programare. Prezentarea normelor de protecția muncii. Familiarizarea cu sistemele de calcul și cu mediul de programare (editare, compilare, execuție, depanare)	2	expunere, verificare/testare	
2. Primele programe în C: structura unui program, tipuri de date, variabile și constante	2	lucrări practice, experimentul	
3. Utilizarea funcțiilor standard de intrare/ieșire (printf, scanf, getchar, putchar)	2	lucrări practice, experimentul	
4. Programe cu operatori, expresii și conversii de tip	2	lucrări practice, experimentul	
5. Programe cu structuri decizionale (if, switch). Testare programare în limbajul C	2	lucrări practice, experimentul, verificare/testare	
6. Programe cu structuri repetitive (while, do-while, for). Algoritmi elementari	2	lucrări practice, experimentul	
7. Lucrul cu vectori și matrice: parcurgere, căutare, sortare	2	lucrări practice, experimentul	
8. Programarea aplicațiilor cu caractere și șiruri de caractere	2	lucrări practice, experimentul	
9. Funcții definite de utilizator. Transmiterea parametrilor. Recursivitate	2	lucrări practice, experimentul	
10. Aplicații cu funcții și pointeri. Testare programare în limbajul C	2	lucrări practice, experimentul, verificare/testare	
11. Alocarea dinamică a memoriei. Depanarea programelor (debugger, detectarea erorilor de memorie)	2	lucrări practice, experimentul	
12. Aplicații ce utilizează tipuri de date agregat: structuri, uniuni, enumerări, tablouri de structuri	2	lucrări practice, experimentul	
13. Aplicații ce utilizează fișiere text și binare	2	lucrări practice, experimentul	
14. Programe modulare: proiecte cu mai multe fișiere sursă, biblioteci, parametri în linia de comandă. Susținerea temelor de casă	2	lucrări practice, experimentul, verificare/testare	
Temele de casă constau în dezvoltarea unor programe de complexitate medie. Etape principale în dezvoltarea temei: - însușirea temei de realizare; - proiectarea logică a produsului program; - programarea; inserarea de comentarii în codul sursă; - punerea la punct a produsului program; - testarea cu date impuse.			
Bibliografie minimală recomandată			

1. Stephen G. Kochan, Programming in C (4th Edition) (Developer's Library), Addison-Wesley Professional, ISBN-13: 978-0321776419, 2014;
2. L. Negrescu - Limbajele C și C++ pentru începători, vol. I, Editura Albastră, Cluj-Napoca, 2012;
3. K. Jamsa, L. Klander - Totul despre C și C++, Editura Teora, 2013;
4. Remus Prodan – pagina web a cursului: http://dev.fiesc.usv.ro/remus;
5. B.W. Kernighan, D.M. Ritchie - Programarea în limbaj C, Editura Teora, 2003;
6. K. N. King, C Programming: A Modern Approach (2nd Edition), W. W. Norton & Company, ISBN-13: 978-0393979503, 2008;
7. Pentiuc St. Gh., Turcu Cristina, Turcu Cornel, Mahalu George, Petrișor Cristinel - Programarea calculatoarelor, Universitatea „Ștefan cel Mare" Suceava, 1995.

Observații:
- La cererea studentului, activitatea pe parcurs poate fi echivalată prin proiecte, pregătirea și participarea la concursuri profesionale sau alte tipuri de activități specifice, cu aprobarea cadrului didactic care conduce lucrările practice;
- Studenții care au mai mult de 50% absențe la activitatea practică pot recupera această activitate cu aprobarea cadrului didactic care conduce lucrările practice.

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Evaluare finală	Cunoștințe: corectitudinea și completitudinea explicării conceptelor limbajului C și a interpretării rezultatelor execuției unor secvențe de cod. Aptitudini: funcționalitatea și corectitudinea unui program C elaborat și testat pe calculator pe baza unei cerințe date. Standard minim: nota 5 la fiecare probă; programul rezolvă cerința de bază și se compilează fără erori.	Examen: probă teoretică (scris sau oral) și probă practică pe calculator (elaborarea, testarea și explicarea unui program C)	50
Evaluare pe parcurs	Aptitudini: corectitudinea programelor realizate la laborator și la cele două testări pe calculator. Responsabilitate și autonomie: prezentarea și argumentarea temei de casă, originalitatea codului și utilizarea declarată a instrumentelor de inteligență artificială. Standard minim: nota 5 la activitatea pe parcurs, cu tema de casă predată și susținută.	Evaluare continuă la laborator; două testări practice pe calculator (programe funcționale); susținerea temei de casă	50

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.
.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
23.09.2026	S.l. dr. ing. Remus Cătălin PRODAN	drd.ing. Adrian-Vasile Catana

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
24.09.2026	conf. dr. ing. Ioan Ungurean

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
24.09.2026	Prof. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
25.09.2026	Prof. dr. ing. Laurentiu- Dan MILICI