

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea ¹ / Departamentul ²	Electronică, Telecomunicații și Tehnologii Informaționale/Calculatoare și Tehnologia Informației
1.3 Domeniul de studii (denumire/cod ³)	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale / 20/20/10/100
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Electronică Aplicată / 20/20/10/100/10/Inginer Electronică Aplicată

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă ⁴	Limbaje de Programare 1 / DF						
2.1b Denumirea disciplinei în limba engleză	Programming Languages 1						
2.2 Titularul activităților de curs	Sl. dr. ing. Petru-Lucian Serafin						
2.3 Titularul activităților aplicative ⁵	Sl. dr. ing. Petru-Lucian Serafin						
2.4 Anul de studii ⁶	1	2.5 Semestrul	1	2.6 Tipul de evaluare	V	2.7 Regimul disciplinei ⁷	DOB

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁸

3.1 Număr de ore asistate integral/săptămână	4 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	2
3.1* Număr total de ore asistate integral/sem.	56 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	28
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	3.14 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			0.5
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1.14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1.5
3.7* Număr total de ore activități neasistate/semestru	44 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			7
		ore studiu individual după manual, suport de curs, bibliografie și notițe			16
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			21
3.8 Total ore/săptămână ⁹	7.14				
3.8* Total ore/semestru	100				
3.9 Număr de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Nu este cazul
4.2 de rezultatele învățării	• Cunoștințe de matematică de nivel Bacalaureat, Competențe digitale de bază

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	• Sală de curs cu proiector și tablă, VS Code free license
5.2 de desfășurare a activităților practice	• Laborator cu 15-20 calculatoare. Mediu de programare C (VSCode/CodeBlocks), acces la rețeaua internă CTI, acces la Campus Virtual cv.upt.ro

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C1. Studentul/absolventul identifică și descrie concepte, principii și metode de bază din matematică, fizică, grafică asistată de calculator, bazele electrotehnicii, limbaje de programare. • C2. Studentul/absolventul explică și interpretează rezultate teoretice și experimentale din matematică, fizică, analiza și sinteza circuitelor, programarea calculatoarelor, și grafica asistată de calculator. • C5. Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la arhitectura sistemelor de calcul, microcontrolere, limbaje și tehnici de programare și modul lor de aplicare în probleme concrete. • C9. Studentul/absolventul descrie, identifică, sumarizează concepte de inginerie electronică, telecomunicații și tehnologii informaționale cum ar fi funcționalitatea, capacitatea de multiplicare și costurile legate de proiectare și modul în care acestea sunt aplicate pentru realizarea proiectelor de inginerie. • C22. Studentul/absolventul explică și corelează concepte fundamentale din științele informatice și domeniul procesării limbajului natural, cu accent pe diferențele dintre traducerea umană și cea asistată de calculator. •
Abilități	• A6. Studentul/absolventul achiziționează și prelucurează date, interpretează rezultate teoretice și experimentale. • A55. Studentul/absolventul utilizează și configurează softuri specializate pentru colectarea, analiza și interpretarea datelor provenite din echipamentele de monitorizare industrială. • A17. Studentul/absolventul specifică cerințe, elaborează programe în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), execută, depanează și interpretează rezultatele programelor realizate în vederea rezolvării unei probleme concrete. • A18. Studentul/absolventul realizează proiecte de complexitate mică/medie care implică utilizarea simultană de componente hardware (microprocesoare sau microcontrolere) și componente software (programele aferente). • A19. Studentul/absolventul elaborează și rezolvă exerciții practice, lucrări de laborator și probleme aplicative, demonstrând capacitatea de integrare a noțiunilor teoretice. • A30. Studentul/absolventul utilizează modelarea, simularea și testarea elementelor procesului într-un mod orientat către probleme în integrarea acestora în timpul dezvoltării •
Responsabilitate și autonomie	• RA2. Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor. • RA4. Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. • RA10. Studentul/absolventul manifestă capacitatea de autoorganizare și de gestionare a timpului de studiu, respectând cerințele și termenele activităților academice. • RA6. Studentul/absolventul lucrează eficient ca membru în echipă sau lider al acesteia. •

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

- Însușirea conceptelor și metodelor de bază privitoare la limbaje și tehnici de programare
- Dobândirea noțiunilor de bază din programare, cu exemplificare în limbajul C
- Dobândirea unei imagini de ansamblu asupra domeniului calculatoarelor și al programării
- Proiectarea și implementarea unor programe C de complexitate mică și medie
- Obținerea unor deprinderi de testare și depanarea a programelor
- Însușirea unui stil de programare corect

8. Conținuturi¹⁰

8.1 Curs	Număr de ore	Metode de predare ¹¹
1 Introducere în programarea calculatoarelor 1.1 Programarea sistemelor de calcul 1.2 Scheme logice 1.3 Limbaje de programare 1.4 Scurt istoric al limbajului C	2	Prelegere susținută de prezentări PPT, conversații, explicații, exemplificări
2 Tipuri de date predefinite. Constante, variabile și expresii 2.1 Funcții C 2.2 Tipuri de date în C 2.3 Operatori și expresii în C 2.4 Conversii de tipuri 2.5 Constante numerice 2.6 Variabile	2 2	
3 Controlul fluxului de execuție. Instrucțiuni	2	

3.1 Structura unui program C 3.2 Secvențierea 3.3 Operații simple de citire/scriere 3.4 Operatori relaționali 3.5 Operatori logici 3.6 Operatorul condițional 3.7 Instrucțiuni condiționale 3.8 Instrucțiuni de ciclare	2	
4 Apeluri de funcții. Recursivitate 4.1 Apeluri de funcții 4.2 Recursivitate 4.3 Recursivitate vs. iterație	2 2	
5 Tablouri 5.1 Variabile locale vs Variabile globale 5.2 Domenii de vizibilitate pentru variabile 5.3 Tablouri unidimensionale 5.4 Tablouri multidimensionale	2	
6 Pointeri. Șiruri de caractere 6.1 Tipul pointer 6.2 Tipul char 6.3 Șiruri de caractere 6.4 Funcții cu șiruri de caractere	2	
7 Pointeri (continuare) 7.1 Tablouri si pointeri 7.2 Aritmetica pointerilor 7.3 Alocarea dinamica a memoriei 7.4 Pointeri la funcții	2	
8 Tipuri definite de utilizator 8.1 Tipul enumerare 8.2 Tipul structura 8.3 Tipul uniune 8.4 Funcții de intrare/ieșire	2 2	
9 Fișiere 9.1 Fișiere text 9.2 Fișiere binare 9.3 Parametri pe linie de comandă 9.4 Funcții cu număr variabil de parametri	2	
10 Reprezentare internă. Operații pe biți 10.1 Reprezentare internă 10.2 Operatori pe biți 10.3 Preprocesorul C	2	
Bibliografie ¹² 1. Kernighan, B. W. & Ritchie, D. "The C programming language - Second edition", 1988 Prentice Hall Software Series 2. Minea, M., Limbaje de programare, materiale de curs http://staff.cs.upt.ro/~marius/curs/lp/index.html 3. Holotescu, C., Limbaje de programare, materiale de curs http://labs.cs.upt.ro/~oose/pmwiki.php/LP/Lectures 4. Năstac, I., Programarea calculatoarelor in limbajul C – Elemente fundamentale, Editura Printech, 2006 5. Năstac, I., Programarea Calculatoarelor, materiale de curs https://www.euroqual.pub.ro/cursuri/programarea-calculatoarelor/		
8.2 Activități aplicative¹³	Număr de ore	Metode de predare
Familiarizare cu mediul de programare. Un prim program C de tip	2	Expunerea temelor cu ajutorul unor materiale auxiliare, exemple, discuții, întrebări, rezolvarea unor aplicații in limbajul C
Tipuri fundamentale.	2	
Operatori matematici si de atribuire	2	
Funcții. Operații simple de citire/scriere	2	
Instrucțiuni de decizie. Funcții recursive	2	
Instrucțiuni de ciclare.	2	
Tablouri (vectori, matrice)	2	
Tipul de date caracter.	2	
Șiruri de caractere	2	
Pointeri-partea 1	2	

Pointeri -partea 2	2	
Tipuri de date definite de utilizator.	2	
Structuri	2	
Operații cu fișiere	2	
Bibliografie ¹⁴ 1. Minea, M., Limbaje de programare, materiale de curs http://staff.cs.upt.ro/~marius/curs/lp/index.html 2. Holotescu, C., Limbaje de programare, materiale de curs http://labs.cs.upt.ro/~oose/pmwiki.php/LP/Lectures 3. Materialele de laborator de Programarea calculatoarelor (OpenCourseWare) 4. Serafin, P-L, Limbaje de Programare – activități practice – http://cv.upt.ro , 2024		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ¹⁵	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Înțelegerea conceptelor prezentate la curs și rezolvarea unor aplicații propuse la fiecare capitol de curs	Verificarea cunoștințelor se efectuează prin examen scris distribuit în trei părți cu subiecte la fiecare parte: Se dau atât subiecte de teorie, prin întrebări de tip grilă, cât și probleme de tipul celor propuse la curs și la laborator în mediul de programare Visual Studio Code, limbajul de programare C. Partea teoretică este evaluată sub forma unui test grilă pe Campus Virtual cu notarea răspunsurilor scris pe formular, iar partea aplicativă este evaluată prin implementarea programelor C compilate pe calculator	0.5
9.5 Activități aplicative	S:		
	L: Implementarea de programe și teste grilă	Studentii sunt evaluați în timpul laboratoarelor prin teste de tip grilă și prin notarea programelor dezvoltate exclusiv în timpul laboratorului. Se pot primi teme de lucru, care vor fi notate cu punctaje suplimentare	0.5
	P¹⁶:		
	Pr:		
9.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁷)			
• Scrierea de cod în limbajul C, program compilabil • Proiectarea, testarea și executarea unui program de complexitate medie: 3-4 funcții, transmiterea de parametri la funcții • Stăpânirea lucrului cu structuri de date: tablouri liniare, șiruri de caractere, operații cu fișiere • Operații de bază în C			

Data completării

4.09.2025

Titular de curs
(semnătura)

Sl. dr. ing. Petru-Lucian Serafin

Titular activități aplicative
(semnătura)

Sl. dr. ing. Petru-Lucian Serafin

Director de departament
(semnătura)

Conf. dr. ing. Mircea Babaita

Data avizării în Consiliul Facultății¹⁸

07.10.2025

Decan
(semnătura)

Prof. dr. ing. Cătălin CĂLEANU