



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie Politehnica București
1.2 Facultatea	Electronică, Telecomunicații și Tehnologia Informației
1.3 Departamentul	Dispozitive, Circuite și Arhitecturi Electronice
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
1.5 Ciclul de studii	Licență
1.6 Specializarea	Electronică aplicată

2. Date despre disciplină

2.1 Denumirea disciplinei (ro)				Programarea calculatoarelor și limbaje de programare 1			
(en)				Computer Programming and Programming Languages 1			
2.2 Titularul activităților de curs				Conf. Dr. Radu Hobincu			
2.3 Titularul activităților de seminar / laborator				Conf. Dr. Radu Hobincu			
2.4 Anul de studiu	1	2.5 Semestrul	I	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei	F	2.9 Codul disciplinei	04.F.01.O.004	2.10 Tipul de notare	Nota		

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2.00	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56.00	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					42
Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					
Tutorat					2
Examinări					0
Alte activități (dacă există):					0
3.7 Total ore studiu individual	44.00				
3.8 Total ore pe semestru	100				
3.9 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Cunoștințe de algebră de bază.
4.2 de rezultate ale învățării	Cunoștințe generale de utilizare a calculatorului.

5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

5.1 Curs	Sală de curs cu proiector și acces Internet.
----------	--



5.2 Seminar/ Laborator/Proiect	Laborator cu calculatoare ce rulează un sistem de operare Linux, ce dispune minim de compilator de C (gcc) și editor de text, de preferat un IDE modern (ex: Jetbrains CLion). Capacitatea laboratorului trebuie să fie de minim 15 locuri și să aibă acces la Internet.
-----------------------------------	--

6. Obiectiv general *(Se referă la intențiile profesorilor pentru studenți, la ceea ce studenții vor fi învățați în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului științific abordat, precum și la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învățământ al specializării studiate etc.)*

Această disciplină este una fundamentală pentru domeniul ETTI și își propune să familiarizeze studenții cu un limbaj de programare de uz general (C). Disciplina este una introductivă, care începe cu introducerea conceptelor generale de paradigmă imperativ-procedurală, continuând cu descrierea sintaxei limbajului. În același timp, disciplina își propune să obișnuiască studentul cu lucrul într-o manieră axată pe implementarea de soluții funcționale, evaluarea făcându-se automat, de către o platformă software configurată prin teste de Intrare-Ieșire.

7. Competențe *(Capacitatea dovedită de a utiliza cunoștințe, aptitudini și abilități personale, sociale și/sau metodologice în situații de muncă sau de studiu și pentru dezvoltarea profesională și personală. Reflectă cerințele angajatorilor.)*

Specifice	C2. Aplicarea, în situații tipice, a metodelor de bază de achiziție și prelucrare ale semnalelor (1/4) C3. Aplicarea cunoștințelor, conceptelor și metodelor de bază privitoare la arhitectura sistemelor de calcul, microprocesoare, microcontrolere, limbaje și tehnici de programare (3/4)
Transversale (generale)	Gândire logică, structurată, capacitatea de analiză și descompunere a unei probleme.

8. Rezultatele învățării *(Sunt enunțuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învățării reflectă realizările studentului și mai puțin intențiile profesorului. Rezultatele învățării informează studenții despre ceea ce se așteaptă de la ei din punct de vedere al performanței, pentru a obține notele și creditele dorite. Sunt definite în termeni concreți, folosind verbe similare exemplelor de mai jos și indică ceea ce se va urmări prin evaluare. Rezultatele învățării vor fi astfel redactate încât să fie evidențiată clar relația față de competențele definite la punctul 7.)*

Cunoștințe	<i>Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau factice.</i> Cunoștințe legate de sintaxa limbajului C și de modul de compilare a programelor în aplicații, precum și de funcționarea programelor în timpul execuției.
Aptitudini	<i>Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente).</i> Aptitudini legate de a analiza o problemă dată în limbaj natural și a dezvolta o soluție proprie pentru rezolvarea problemei, precum și capacitatea de a implementa această soluție în limbajul C.



Responsabilitate
și autonomie

Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale. Capacitatea de a căuta răspunsuri și a se documenta online legat de sintaxa și API-ul limbajului C, precum și de a căuta soluții la probleme punctuale.

9. Metode de predare (Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenților la stabilirea propriului parcurs de învățare, cum se identifică eventualele rămăneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)

După prezentarea teoretică a noțiunilor necesare, cu ajutorul proiecteurului și a unui calculator, se exemplifică practic prin scrierea de mici programe și analiza rezultatelor. Se cere constant feedback studenților în legătură cu calitatea explicațiilor și pentru sugestii de implementare a exemplelor. Noțiunile teoretice sunt prezentate cu ajutorul unei tablete și Stylus care proiectează informația pe ecran, urmând ca la sfârșitul cursului, notițele să fie publicate pe Moodle. În laborator studenții beneficiază de un calculator și un set de exerciții pe care le rezolvă individual cu ajutorul cadrului didactic și a platformelor de laborator.

10. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore
1	Sisteme digitale. Ciclul de viață al unei aplicații (Software Development Life-Cycle - SDLC) - Arhitectură și Design; Noțiuni elementare despre compilator, limbaj și medii de dezvoltare	1
2	Variabila și operatorul de atribuire – lvalue și rvalue	1
3	Tipuri de date numerice – int și double; Fluxurile unui proces; Formate pentru citirea și scrierea datelor (printf & scanf); Operatori aritmetici	3
4	Baze de numerație – binar, decimal și hexazecimal; Stocarea numerelor întregi negative – complement față de doi; Tipuri de date întregi de dimensiune fixă; Operatori pe biți; Stocarea numerelor în virgulă mobilă – Standardul IEEE 754	3
5	Structuri condiționale: if-else, switch-case și operatorul condițional "?"; Operatori logici și evaluarea expresiilor logice	2
6	Operatorul de conversie; Expresii repetitive - while, do-while, for. Operatorul de secvențiere (.)	2
7	Vectori uni și multi-dimensionali; Stocarea caracterelor – codurile ASCII și Unicode; Șiruri de caractere	4
8	Funcții, variabile locale și globale; Pass-by-value vs. Pass-by-reference; Modelul de execuție al unui program. Funcții recursive. Funcții tail-recursive	4
9	Structuri: struct, enum și union	2
10	Pointeri și aritmetica pointerilor; Alocarea dinamică de memorie; Lucrul cu fișiere	6
	Total:	



Bibliografie:

1. Radu HOBINCU, Călin BÎRĂ, Programarea Calculatoarelor, suport de curs electronic, <https://curs.upb.ro/>
2. Greg Perry, Dean Miller, "C Programming Absolute Beginner's Guide, 3rd Edition", Que Publishing, ISBN: 0789751984, 2013
3. <http://en.cppreference.com/w/c> - Descrierea API-ului limbajului C
4. <https://gcc.gnu.org/onlinedocs/gcc/> - Documentația compilatorului gcc

LABORATOR

Nr. crt.	Conținutul	Nr. ore
1	Linux – comenzi de bază în Bash	2
2	GNU Compiler Collection, GNU Make	2
3	Fluxurile unui proces	2
4	Tipuri de date și operatori numerici	2
5	Expresii condiționale	2
6	Expresii repetitive	2
7	Vectori uni și multi-dimensionali	2
8	Șiruri de caractere și funcții pentru manipularea acestora	2
9	Funcții	2
10	Funcții recursive; GNU Debugger - gdb	2
11	Structuri (struct, union și enum)	2
12	Pointerii; valgrind	2
13	Lucrul cu fișiere	2
14	Colocviu final de laborator	2
	Total:	

Bibliografie:

Platforme de laborator - [https://wiki.dcae.pub.ro/index.php/Programarea_Calculatoarelor_\(laborator\)](https://wiki.dcae.pub.ro/index.php/Programarea_Calculatoarelor_(laborator))

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	Însușirea noțiunilor teoretice și a sintaxei limbajului C pe parcursul semestrului.	Zece teme de casă de complexitate medie spre ridicată, a căror soluții se testează automat pe platforma Moodle.	10
	Însușirea noțiunilor teoretice și a sintaxei limbajului C în ansamblu.	Examen final, susținut pe platforma Moodle.	40



11.5 Seminar/laborator/proiect	Gradul în care studentul și-a îndeplinit obligațiile de studiu individual.	Test teoretic la începutul fiecărui laborator.	5
	Capacitatea studentului de a folosi noțiunile acumulate pentru a rezolva probleme practice.	Rezolvarea de probleme de programare în timpul fiecărui laborator.	15
	Capacitatea de a abstractiza o problemă reală și a implementa o soluție în limbajul C.	Colocviu de laborator, susținut pe platforma Moodle, în care se cere rezolvarea individual a unei probleme simple, similare cu cele întâlnite în laborator, cu cerința specifică ca aceasta să fie implementată corect și complet pentru acordarea punctajului.	30
11.6 Condiții de promovare			
1. Obținerea a 50% din punctajul total. 2. Obținerea a 50% din punctajul aferent activității de laborator.			

12. Coroborarea conținutului disciplinei cu așteptările reprezentanților angajatorilor și asociațiilor profesionale reprezentative din domeniul aferent programului, precum și cu stadiul actual al cunoașterii în domeniul științific abordat și practicile în instituții de învățământ superior din Spațiul European al Învățământului Superior (SEİS)

Programarea calculatoarelor este primul pas în vastul domeniu al ingineriei software. În ziua de astăzi, când inclusiv dezvoltarea circuitelor și sistemelor se face prin limbaje dedicate de descriere hardware, este de neconceput pentru un absolvent de facultate tehnică să nu aibă cunoștințe puternice de programare. Această nevoie acută de experiență în programare a fost subliniată în mod repetat de reprezentanți ai industriei IT din România în relațiile cu mediul academic. Prin urmare, Facultatea de Electronică din cadrul UPB a luat chiar măsuri suplimentare pentru ameliorarea nivelului acestor cunoștințe în rândul absolvenților noștri.

Data completării

Titular de curs

Titular(i) de aplicații

25.09.2025

Conf. Dr. Radu
Hobincu

As. Drd. Ing. Costin-Emanuel VASILE

As. Drd. Ing. Andrei-Alexandru
ULMĂMEI



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



As. Drd. Ing. Alexandra ENESCU

As. Drd. Ing. Alexandru GUZU

Data avizării în departament

Director de departament

Prof. Dr. Ing. Claudiu DAN

Data aprobării în Consiliul
Facultății

Decan

26.09.2025

Prof. Dr. Mihnea Udrea