Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti
Facultatea de Electronică, Telecomunicaţii şi
Tehnologia Informaţiei

FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	**Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti**
1.2 Facultatea	**Electronică, Telecomunicaţii şi Tehnologia Informaţiei**
1.3 Departamentul	**Dispozitive, Circuite şi Arhitecturi Electronice**
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicaţii şi Tehnologii Informaţionale
1.5 Ciclul de studii	Licenţă
1.6 Specializarea	Electronică aplicată

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)				Programarea calculatoarelor şi limbaje de programare 1 Computer Programming and Programming Languages 1			
2.2 Titularul activităţilor de curs				Conf. Dr. Radu Hobincu			
2.3 Titularul activităţilor 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 activităţilor 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 învăţământ	56.00	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuţia fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie şi notiţe Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii şi eseuri					42
Tutorat					2
Examinări					0
Alte activităţi (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. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Cunoştinţe de algebră de bază.
4.2 de rezultate ale învăţării	Cunoştinţe generale de utilizare a calculatorului.

5. Condiţii necesare pentru desfăşurarea optimă a activităţilor didactice (acolo unde este cazul)

5.1 Curs	Sală de curs cu proiector şi 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) şi editor de text, de preferat un IDE modern (ex: Jetbrains CLion). Capacitatea laboratorului trebuie să fie de minim 15 locuri şi să aibă acces la Internet.

6. Obiectiv general *(Se referă la intenţiile profesorilor pentru studenţi, la ceea ce studenţii vor fi învăţaţi în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului ştiinţific abordat, precum şi 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 învăţământ al specializării studiate etc.)*

Această disciplină este una fundamentală pentru domeniul ETTI şi îşi propune să familiarizeze studenţii 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 acelaşi timp, disciplina îşi propune să obişnuiască studentul cu lucrul într-o manieră axată pe implementarea de soluţii funcţionale, evaluarea făcându-se automat, de către o platformă software configurată prin teste de Intrare-Ieşire.

7. Competenţe *(Capacitatea dovedită de a utiliza cunoştinţe, aptitudini şi abilităţi personale, sociale şi/sau metodologice în situaţii de muncă sau de studiu şi pentru dezvoltarea profesională şi personală. Reflectă cerinţele angajatorilor.)*

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

8. Rezultatele învăţării *(Sunt enunţuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învăţării reflectă realizările studentului şi mai puţin intenţiile profesorului. Rezultatele învăţării informează studenţii despre ceea ce se aşteaptă de la ei din punct de vedere al performanţei, pentru a obţine notele şi creditele dorite. Sunt definite în termeni concreţi, folosind verbe similare exemplelor de mai jos şi indică ceea ce se va urmări prin evaluare. Rezultatele învăţării vor fi astfel redactate încât să fie evidenţiată clar relaţia faţă de competenţele definite la punctul 7.)*

Cunoştinţe	*Rezultatul asimilării de informaţii prin învăţare. Cunoştinţele reprezintă ansamblul de fapte, principii, teorii şi practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice şi/sau faptice.* Cunoştinţe legate de sintaxa limbajului C şi de modul de compilare a programelor în aplicaţii, precum şi de funcţionarea programelor în timpul execuţiei.
Aptitudini	*Capacitatea de a aplica cunoştinţe şi de a utiliza know-how pentru a duce la îndeplinire sarcini şi a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive şi creative) sau practice (implicând dexteritate manuală şi utilizarea de metode, materiale, unelte şi instrumente).* Aptitudini legate de a analiza o problem dată în limbaj natural şi a dezvolta o soluţie proprie pentru rezolvarea problemei, precum şi capacitatea de a implementa această soluţie 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 şi a se documenta online legat de sintaxa şi API-ul limbajului C, precum şi de a căuta soluţii 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 rămâneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)*

După prezentarea teoretică a noţiunilor necesare, cu ajutorul proiectorului şi a unui calculator, se exemplifică practic prin scrierea de mici programe şi analiza rezultatelor. Se cere constant feedback studenţilor în legătură cu calitatea explicaţiilor şi pentru sugestii de implementare a exemplelor. Noţiunile teoretice sunt prezentate cu ajutorul unei tablete şi Stylus care proiectează informaţia pe ecran, urmând ca la sfârşitul cursului, notiţele să fie publicate pe Moodle. În laborator studenţii beneficiază de un calculator şi un set de exerciţii pe care le rezolvă individual cu ajutorul cadrului didactic şi a platformelor de laborator.

10. Conţinuturi

CURS		
Capitolul	**Conţinutul**	**Nr. ore**
1	Sisteme digitale. Ciclul de viaţă al unei aplicaţii (Software Development Life-Cycle - SDLC) - Arhitectură şi Design; Noţiuni elementare despre compilator, limbaj şi medii de dezvoltare	1
2	Variabila şi operatorul de atribuire – lvalue şi rvalue	1
3	Tipuri de date numerice – int şi double; Fluxurile unui proces; Formate pentru citirea şi scrierea datelor (printf & scanf); Operatori aritmetici	3
4	Baze de numeraţie – binar, decimal şi hexazecimal; Stocarea numerelor întregi negative – complement faţă de doi; Tipuri de date întregi de dimensiune fixă; Operatori pe biţi; Stocarea numerelor în virgulă mobilă – Standardul IEEE 754	3
5	Structuri condiţionale: if-else, switch-case şi operatorul condiţional "?"; Operatori logici şi evaluarea expresiilor logice	2
6	Operatorul de conversie; Expresii repetitive - while, do-while, for. Operatorul de secvenţiere (,)	2
7	Vectori uni şi multi-dimensionali; Stocarea caracterelor – codurile ASCII şi Unicode; Şiruri de caractere	4
8	Funcţii, variabile locale şi globale; Pass-by-value vs. Pass-by-reference; Modelul de execuţie al unui program. Funcţii recursive. Funcţii tail-recursive	4
9	Structuri: struct, enum şi union	2
10	Pointeri şi aritmetica pointerilor; Alocarea dinamică de memorie; Lucrul cu fişiere	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/ - Documentaţia compilatorului gcc

LABORATOR

Nr. crt.	Conţinutul	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 şi operatori numerici	2
5	Expresii condiţionale	2
6	Expresii repetitive	2
7	Vectori uni şi multi-dimensionali	2
8	Şiruri de caractere şi funcţii pentru manipularea acestora	2
9	Funcţii	2
10	Funcţii recursive; GNU Debugger - gdb	2
11	Structuri (struct, union şi enum)	2
12	Pointeri; valgrind	2
13	Lucrul cu fişiere	2
14	Colocviu final de laborator	2
	Total:	

Bibliografie:
Platforme de 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	Însuşirea noţiunilor teoretice şi a sintaxei limbajului C pe parcursul semestrului.	Zece teme de casă de complexitate medie spre ridicată, a căror soluţii se testează automat pe platforma Moodle.	10
	Însuşirea noţiunilor teoretice şi a sintaxei limbajului C în ansamblu.	Examen final, susţinut pe platforma Moodle.	40

11.5 Seminar/laborator/proiect	Gradul în care studentul şi-a îndeplinit obligaţiile de studiu individual.	Test teoretic la începutul fiecărui laborator.	5
	Capacitatea studentului de a folosi noţiunile acumulate pentru a rezolva probleme practice.	Rezolvarea de probleme de programare în timpul fiecărui laborator.	15
	Capacitatea de a abstractiza o problemă reală şi a implementa o soluţie 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 Condiţii de promovare			
1. Obținerea a 50% din punctajul total. 2. Obținerea a 50% din punctajul aferent activității de laborator.			

12. Coroborarea conţinutului disciplinei cu aşteptările reprezentanţilor angajatorilor şi asociaţiilor profesionale reprezentative din domeniul aferent programului, precum şi cu stadiul actual al cunoaşterii în domeniul ştiinţific abordat şi practicile în instituţii de învăţământ superior din Spaţiul European al Învăţământului Superior (SEÎS)

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 reprezentați ai industriei IT din Romania 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 aplicaţii
21.09.2026	Conf. Dr. Radu Hobincu	As. Drd. Ing. Costin-Emanuel VASILE
		As. Drd. Ing. Andrei-Alexandru ULMĂMEI

Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti
Facultatea de Electronică, Telecomunicaţii şi
Tehnologia Informaţiei

As. Drd. Ing. Alexandra ENESCU

As. Drd. Ing. Alexandru GUZU

Data avizării în departament	Director de departament
	Prof. Dr. Ing. Claudius DAN
Data aprobării în Consiliul Facultăţii	Decan
22.09.2026	Prof. Dr. Mihnea Udrea