



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie POLITEHNICA din București
1.2 Facultatea	Științe Aplicate
1.3 Departamentul	
1.4 Domeniul de studii universitare	Științe ingineresti aplicate
1.5 Programul de studii universitare	Matematică și informatică aplicată în inginerie
1.6 Ciclul de studii universitare	Licență
1.7 Limba de predare	Română
1.8 Locația geografică de desfășurare a studiilor	București

2. Date despre disciplină

2.1 Denumirea disciplinei/ Course title (ro) (en)	Programarea calculatoarelor si limbaje de programare 1 Computers programming and programming languages 1						
2.2 Titularul/ii activităților de curs	Prof. dr. ing. Andreea Udrea						
2.3 Titularul/ii activităților de seminar / laborator/proiect	as. drd. Miruna Vlasceanu, as.drd. Ioana Stefan						
2.4 Anul de studiu	1	2.5 Semestrul/	I	2.6. Tipul de evaluare	E	2.7 Statutul disciplinei	Ob
2.8 Categoria formativă	DF	2.9 Codul disciplinei					

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

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator/proiect	2
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs/	28	3.6 seminar/laborator/proiect	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					40
Tutorat					0
Examinări					4
Alte activități (dacă există):					X
3.7 Total ore studiu individual	44				
3.8 Total ore pe semestru	100				
3.9 Numărul de credite	4				



4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de rezultate ale învățării	

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

5.1 de desfășurare a cursului	Cursul se va desfășura într-o sală dotată cu videoproiector și computer.
5.2 de desfășurare a seminarului/laboratorului/proiectului	Laboratorul se va desfășura într-o sală cu dotare specifică, care trebuie să includă: PC-uri cu conexiune la internet și IDE C

6. Obiectiv general

Disciplina „Programarea calculatoarelor și limbaje de programare 1” are ca obiectiv general formarea capacității studenților de a înțelege și aplica conceptele fundamentale ale programării structurate, utilizând limbajul C ca mediu de lucru. Prin parcurgerea acestei discipline, studenții vor dobândi cunoștințe teoretice și deprinderi practice necesare pentru dezvoltarea de algoritmi, scrierea, testarea și optimizarea programelor informatice. Totodată, se urmărește formarea unei gândiri logice și algoritmice, înțelegerii modului de organizare a memoriei și a mecanismelor interne ale procesării, precum și dezvoltarea abilității de a selecta și utiliza eficient structuri de date și tehnici de programare. Un alt obiectiv major este familiarizarea studenților cu diferențele dintre limbajele compilate și cele interpretate, respectiv cu principiile de bază ale organizării spațiului de adresă al unui proces, aspecte esențiale pentru înțelegerea arhitecturii și funcționării programelor pe calculator. Disciplina sprijină astfel dezvoltarea competențelor necesare pentru continuarea studiilor în domenii avansate ale informaticii și pentru rezolvarea unor probleme concrete prin mijloace informatice.

○ **7. Rezultatele învățării**

Cunoștințe	Identifică și descrie concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică. Explică și interpretează rezultate teoretice și experimentale din matematică, fizică, chimie, economie, desen tehnic și informatică. Descrie, identifică, sumarizează, prelucrează, concepte și noțiuni elementare referitoare la principii, legi, noțiuni de bază din domeniul științelor fundamentale, analizează și prelucrează modul lor de aplicare în probleme concrete din programului de studii. Descrie, identifică, sumarizează concepte și noțiuni ingineresti și modul lor de aplicare în probleme concrete de uz general specifice programului de studii.
------------	--



Abilități	Operează cu concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică. Utilizează metode fundamentale, explică, utilizează, combină, analizează, noțiuni fundamentale, din domeniul științelor fundamentale pentru a implementa, modela și simula fenomene și sisteme specifice domeniului studiat. Utilizează metode și instrumente specifice pentru studiul, analiza, sinteza și realizarea sistemelor și echipamentelor specifice programului de studii. Proiectează, măsoară, evaluează performanțele, diagnostichează și depanează blocuri funcționale de complexitate mică/medie, folosind medii de modelare și simulare dedicate. Proiectează experimente și sisteme ingineresti funcționale de complexitate mică/medie specifice.
Responsabilitate și autonomie	Aplică valorile eticii și deontologiei profesiei de inginer. Interpretează legi și principii ale științelor fundamentale ce stau la baza fenomenelor și aparatelor din domeniul de studii. Practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor. Comunică eficient despre activitățile de inginerie cu o gamă largă de public. Este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. Selectează și analizează surse bibliografice. Produce software și îl adaptează continuu la noile tehnologii și cerințe de piață. Demonstrează autonomie în învățare Își asumă în mod responsabil sarcinile profesionale și respectă normele de etică și deontologie profesională. Lucrează eficient ca membru în echipă sau lider al acesteia.

8. Metode de predare

Predarea acestei discipline va fi centrată pe student, având ca scop dezvoltarea treptată a competențelor de programare, printr-un echilibru între transmiterea de cunoștințe teoretice și activități practice aplicative.

Procesul didactic va utiliza atât metode expositive (prelegerea, expunerea de concepte fundamentale de programare și structuri de date), cât și metode interactive și practice, precum: demonstrații de cod, rezolvarea de probleme și exerciții individuale. Pentru situațiile în care se constată dificultăți de înțelegere sau rămânări în urmă, se vor realiza sesiuni suplimentare de exerciții și exemple explicate pas cu pas.

La începutul fiecărui curs se va face recapitularea noțiunilor de programare din capitolele anterioare, cu accent pe conceptele predate anterior, pentru a asigura continuitatea în învățare. Prezentările vor fi susținute cu materiale vizuale (scheme logice, diagrame de flux, fragmente de cod comentat) pentru a facilita înțelegerea abstractizărilor teoretice.

Activitățile practice vor avea un rol central, vizând dezvoltarea gândirii algoritmice și a capacității de a transpune o problemă într-o soluție programatică. Studenții vor participa la exerciții de rezolvare de probleme, la scrierea și testarea de cod, precum și la proiecte individuale sau de echipă. Lucrul colaborativ va contribui la dezvoltarea abilităților de comunicare tehnică și de cooperare, esențiale pentru activitatea profesională în domeniul informatic.

De asemenea, se va încuraja oferirea și primirea de feedback constructiv pentru a sprijini progresul individual și adaptarea continuă a metodei de predare la nevoile reale ale grupei.



9. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore
I	Algoritmi – definiție, proprietăți, reprezentare și exemple. Date de intrare și ieșire. Tipuri fundamentale de date. Variabile. Operatori.	2
II	Instrucțiuni: simple, compuse, if, switch, break, while, do-while, for, continue.	2
III	Tipuri de date masive (tablouri) – vectori, matrice. Metode de căutare și sortare.	2
IV	Siruri de caractere	2
V	Tipul pointer și legătura cu tablourile	2
VI	Funcții definite de către utilizator și recursivitate. Transmiterea parametrilor în funcții prin valoare. Transmiterea parametrilor în funcții prin referință.	2
VII	Spațiul virtual de adresă al unui proces. Tipuri de variabile.	2
VIII	Alocare dinamică a memoriei.	2
IX	Directive preprocesor. Limbaje compilate vs. limbaje interpretate.	2
X	Tipuri de date structurate: structuri și câmpuri de biți	2
XI	Tipuri de date structurate: uniuni, enumerări	2
XII	Fisiere text și binare	2
XIII	Notiuni legate de organizarea aplicațiilor complexe. Exemple.	2
XIV	Notiuni generale despre complexitatea algoritmilor și structuri de date.	2
Total:		28
Bibliografie:		
1.		
2. E. Cerchez, M. Serban, Programarea în limbajul C/C++ pentru liceu, vol I, Editura: Polirom, 2004		
3. E. Cerchez, M. Serban, Programarea în limbajul C/C++ pentru liceu, vol II, Editura: Polirom, 2005		
4. E. Cerchez, M. Serban, Programarea în limbajul C/C++ pentru liceu, vol III, Editura: Polirom, 2006		
5. T. Cormen et. al., Introduction to Algorithms, 3rd Edition The MIT Press, 20		
6. N. Karumanchi, Data Structures and Algorithms Made Easy: 700 Data Structure and Algorithmic Puzzles (1st. ed.). CreateSpace, Scotts Valley, CA., . 2011.		

LABORATOR/ SEMINAR/PROIECT		
Nr. crt.	Conținutul	Nr. ore
1.	Algoritmi – definiție, proprietăți, reprezentare și exemple. Date de intrare și ieșire. Tipuri fundamentale de date. Variabile. Operatori.	2
2.	Instrucțiuni: simple, compuse, if, switch, break, while, do-while, for, continue.	2
3.	Tipuri de date masive (tablouri) – vectori, matrice. Metode de căutare și sortare.	2
4.	Siruri de caractere	2
5.	Tipul pointer și legătura cu tablourile	2
6.	Funcții definite de către utilizator și recursivitate. Transmiterea parametrilor în funcții prin valoare. Transmiterea parametrilor în funcții prin referință.	2
7.	Spațiul virtual de adresă al unui proces. Tipuri de variabile.	2
8.	Alocare dinamică a memoriei.	2
9.	Directive preprocesor. Limbaje compilate vs. limbaje interpretate.	2



10.	Tipuri de date structurate: structuri si campuri de biti	2
11.	Tipuri de date structurate: uniuni, enumerari	2
12.	Fisiere text si binare	2
13.	Notiuni legate de organizarea aplicatiilor complexe. Exemple.	2
14.	Notiuni generale despre complexitatea algoritmilor si structuri de date.	2
Total:		28

Bibliografie:

- 1.
2. E. Cerchez, M. Serban, Programarea in limbajul C/C++ pentru liceu, vol I, Editura: Polirom, 2004
3. E. Cerchez, M. Serban, Programarea in limbajul C/C++ pentru liceu, vol II, Editura: Polirom, 2005
4. E. Cerchez, M. Serban, Programarea in limbajul C/C++ pentru liceu, vol III, Editura: Polirom, 2006

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Evaluare elemente practice – probleme	Verificari intermediare	20%
	Evaluare elemente practice si teoretice – probleme (25%) si grile(25%)	Examen	50%
10.5 Seminar/laborator/proiect	Utilizare noțiuni teoretice în aplicații practice	Rezolvare de probleme (cu notare la fiecare laborator)	30%
10.6 Condiții de promovare			
Obținerea a 50% din punctajul total.			

Data completării

Titular de curs

Titular(ii) de aplicații

Udrea Andreea

Vlasceanu Miruna, Stefan Ioana

Data avizării în departament

Director de departament

Data aprobării în Consiliul Facultății

Decan
Alina Claudia PETRESCU-NIȚĂ

26.09.2025