



**Universitatea Națională de Știință și
Tehnologie POLITEHNICA București
Facultatea Știința și Ingineria Materialelor**



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie POLITEHNICA din București
1.2 Facultatea	Știința și ingineria materialelor
1.3 Departamentul	Știința materialelor metalice, metalurgie fizică
1.4 Domeniul de studii universitare	Științe inginerești aplicate
1.5 Programul de studii universitare	Inginerie medicală
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	LIMBAJE ORIENTATE PE OBIECTE OBJECT-ORIENTED LANGUAGES						
2.2 Titularul/ii activităților de curs	S.I.dr.ing.ec. Daniela Alina NECȘULESCU						
2.3 Titularul/ii activităților de seminar / laborator/proiect	S.I.dr.ing.ec. Daniela Alina NECȘULESCU						
2.4 Anul de studiu	3	2.5 Semestrul	I	2.6. Tipul de evaluare	V	2.7 Statutul disciplinei	Op
2.8 Categoria formativă	D	2.9 Codul disciplinei	10.D.05.A.002				

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 seminare/ laboratoare/proiecte, teme, referate, portofolii și eseuri					61
Tutorat					4
Examinări					4
Alte activități (dacă există):					0
3.7 Total ore studiu individual	69				
3.8 Total ore pe semestru	125 ¹				
3.9 Numărul de credite	5				

¹ Se va calcula ținând cont că se acordă un credit pentru volumul de muncă care îi revine unui student cu frecvență la zi pentru a echivala 25 de ore de pregătire pentru dobândirea rezultatelor învățării.



**Universitatea Națională de Știință și
Tehnologie POLITEHNICA București
Facultatea Știința și Ingineria Materialelor**



4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele Programării, Structuri de date și algoritmi
4.2 de rezultate ale învățării	- Abilități medii de programare într-un limbaj de programare de nivel înalt - Implementarea algoritmilor în limbajul C++

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

5.1 de desfășurare a cursului	Sală dotată cu tablă, videoproiector, calculator
5.2 de desfășurare a seminarului/laboratorului/proiectului	Laborator dotat cu tabla și calculatoare prevăzute cu software specializat desfășurării aplicațiilor, limbajul C++

6. Obiectiv general

Programarea orientată obiect are drept obiectiv înțelegerea și însușirea noțiunilor fundamentale, conceptelor și exemplelor privind programarea orientată obiect.

7. Rezultatele învățării

- Realizarea și analiza unor algoritmi specifici programării orientate obiect
- Familiarizarea studenților cu tehnici de bază specifice programării orientate obiect
- Realizarea și testarea unor aplicații în limbajul C++
- Cunoașterea facilităților unui mediu de programare modern
- Lucrul în echipă.

Cunoștințe	Evidențiază cele mai importante diferențe între proiectarea imperativă tradițională și proiectarea orientată pe obiecte. Definește noțiuni specifice programării orientate obiect. Describe noțiuni specifice programării orientate obiect. Explica structurile de tip clasă ca fiind componente fundamentale, modulare. Rezolva diverse aplicații într-o manieră orientată pe obiecte. Înțelege rolul moștenirii, polimorfismului, legării dinamice și a structurilor generice în dezvoltarea unor programe reutilizabile. Evidențiază structura unui program dezvoltat într-un limbaj de programare. Concepe și testează programe C++ de diverse dimensiuni
------------	---



**Universitatea Națională de Știință și
Tehnologie POLITEHNICA București
Facultatea Știința și Ingineria Materialelor**



Abilități	• Utilizarea gândirii analitice. • Dezvoltarea cunoștințelor prin studiu individual. • Întelegerea operațiilor de bază în programarea orientată obiect. • Verifică experimental soluțiile identificate. • Rezolvă aplicații practice. • Identifică soluții și elaborează planuri de rezolvare a problemelor matematice. • Formulează concluzii la programele realizate. • Testează soluțiile identificate/modurile de rezolvare.
Responsabilitate și autonomie	• Selectează surse bibliografice potrivite și le analizează. • Respectă principiile de etică academică, citând corect sursele bibliografice utilizate. • Demonstrează receptivitate pentru contexte noi de învățare. • Manifestă colaborare cu ceilalți colegi și cadre didactice în desfășurarea activităților didactice • Demonstrează autonomie în organizarea situației/contextului de învățare sau a situației problemă de rezolvat • Manifestă responsabilitate socială prin implicarea activă în viața socială studentescă/implicare în evenimentele din comunitatea academică • Promovează/contribuie prin soluții noi, aferente domeniului de specialitate pentru a îmbunătăți calitatea vieții sociale. • Conștientizează valoarea contribuției sale în domeniul ingineriei la identificarea de soluții viabile/sustenabile care să rezolve probleme din viața socială și economică (responsabilitate socială).

8. Metode de predare

Pornindu-se de la analiza caracteristicilor de învățare ale studenților și de la nevoile lor specifice, procesul de predare va include metode de predare atât expositive (prelegerea, expunerea), cât și interactive, bazate pe modele de învățare având la bază modelarea matematica și schemele logice dar și pe metode bazate pe activitățile practice și rezolvarea de probleme.

În activitatea de predare vor fi utilizate prezentări Power Point sau suport digital sub forma de text care vor fi puse la dispoziția studenților. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noțiunilor parcurse la ultimul curs.

Prezentările utilizează concepte ale limbajului de programare, algoritmi și scheme logice, astfel încât informațiile prezentate să fie ușor de înțeles și asimilat.

Această disciplină acoperă informații și activități practice menite să-i sprijine pe studenți în eforturile de învățare și de dezvoltare a unor aplicații informatice într-un climat optim de colaborare și comunicare, favorabil învățării prin descoperire și studiu individual.

Se va avea în vedere exersarea abilităților de ascultare activă și de utilizare a gândirii abstracte în realizarea proiectelor informatice aplicative.

Se va exersa abilitatea de lucru în echipă pentru rezolvarea diferitelor sarcini de învățare.



**Universitatea Națională de Știință și
Tehnologie POLITEHNICA București**
Facultatea Știința și Ingineria Materialelor



9. Conținuturi

CURS		
Capit olul	Conținutul	Nr. ore
I	Elemente de bază ale limbajului C&C++	2
II	Principiile programării orientate obiect în C++	2
III	Clase și obiecte	2
IV	Constructor / destructor	2
V	Date și funcții membre statice	2
VI	Clase și funcții friend	2
VII	Supraîncărcarea operatorilor	2
VIII	Clase: containere și iteratori	2
IX	Moștenirea în programarea orientată obiect	2
X	Polimorfismul - clase și funcții virtuale	2
XI	Clase abstracte, clase friend, clase parametrizate	2
XII	Biblioteca STL	2
XIII	Tratarea excepțiilor în limbajul C++	2
XIV	Șabloane de proiectare	2
	Total:	28
Bibliografie:		
1. L. Negrescu, <i>Limbajul C++, Ed. Albastra, Cluj-Napoca 1996</i>		
2. R. Andonie, I. Garbacea, <i>Algoritmi fundamentali. O perspectiva C++, Editura Libris,</i>		
3. Alexandrescu, <i>Programarea moderna in C++. Programare generica si modele de</i>		
<i>proiectare aplicate, Editura Teora, 2002</i>		
4. E. Horowitz, S. Sahni, D. Mehta, <i>Fundamentals of Data Structures in C++, Computer</i>		
<i>Science Press, Oxford, 1995</i>		
5. Herbert Schildt: <i>C++ Manual Complet, Bucuresti, Ed. Teora 2000</i>		
6. Dan Roman, <i>Ingineria programarii obiectuale, Editura Albastra, Cluj Napoca, 1996.</i>		

LABORATOR/ SEMINAR/PROIECT		
Nr. crt	Conținutul	Nr. ore
1.	Operatori, tipuri de date, instrucțiuni, funcții	2
2.	Clase și obiecte, referințe și pointeri la obiecte	2
3.	Specificarea, proiectarea și implementarea unei clase în C++	2
4.	Exemple de constructor / destructor	2
5.	Tipuri de date și funcții membre statice	2
6.	Categorii de clase și funcții friend	2
7.	Metode de supraîncărcare a operatorilor	2
8.	Exemple de implementare de clase container și iteratori	2



**Universitatea Națională de Știință și
Tehnologie POLITEHNICA București
Facultatea Știința și Ingineria Materialelor**



9.	Noțiunea de moștenire și descrierea ierarhiilor de clase	2
10.	Funcții virtuale și implementarea polimorfismului	2
11.	Exemple de clase abstracte, clase friend, clase parametrizate	2
12.	Folosirea algoritmilor din STL	2
13.	Tratarea excepțiilor în C++	2
14.	Tipuri de șabloane de proiectare	2
Total:		28

Bibliografie:

1. L. Negrescu, *Limbajul C++, Ed. Albastra, Cluj-Napoca 1996*
2. R. Andonie, I. Garbacea, *Algoritmi fundamentali. O perspectiva C++, Editura Libris,*
3. Alexandrescu, *Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002*
4. E. Horowitz, S. Sahni, D. Mehta, *Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995*
5. Herbert Schildt: *C++ Manual Complet, Bucuresti, Ed. Teora 2000*
6. Dan Roman, *Ingineria programarii obiectuale, Editura Albastra, Cluj Napoca, 1996.*

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- Claritatea, coerența și concizia expunerii; - Gradul de acoperire al problematicei solicitate de subiecte; - Utilizarea corectă a termenilor, noțiunilor și tehnologiile specifice cursului.	Examen final 40 p	40%
10.5 Seminar/laborator/proiect	- Participarea activă la laboratoare; - Aplicarea metodelor specifice de rezolvare pentru problema dată; - Utilizarea coerentă și fluentă a termenilor specifici; - Capacitatea de exemplificare.	Prezența la lab. 10 p Activitatea la laborator 20 p Tema de casa 10 p Examen parțial 20 p	60%
10.6 Condiții de promovare			
• prezența la laboratoare; • predarea temei de casă;			



**Universitatea Națională de Știință și
Tehnologie POLITEHNICA București
Facultatea Știința și Ingineria Materialelor**



- obținerea a 30 % din punctajul acumulat în timpul semestrului;
- obținerea a 50 % din punctajul verificării finale (examenului)
- utilizarea conceptelor și instrumentelor din domeniul informaticii.

Obs:

- Ponderea rezultatelor obținute la activitățile din timpul semestrului / notă examen final este 60%: 40%
- În timpul semestrului se poate organiza examen parțial;
- În cazul în care studentul participă la conferințe (studentești, locale, naționale, internaționale) sau concursuri (naționale, internaționale) care au ca tematică această disciplină, acesta va putea beneficia de puncte suplimentare sau de echivalarea unor teme de casa și/sau lucrări și/sau prezență, în funcție de rezultatele obținute;
- La lucrările scrise studenții nu au voie să folosească echipamente electronice, cu excepția calculatoarelor științifice simple.

Data completării

Titular de curs

Titular(ii) de aplicații

02.09.2026

S.l.dr.ing.ec. Daniela Alina
NECȘULESCU

S.l.dr.ing.ec. Daniela Alina
NECȘULESCU

Data avizării în
departament

Director de departament

24.09.2025

Vasile Iulian ANTONIAC

Data aprobării în Consiliul
Facultății

Decan

24.09.2025

Radu ȘTEFĂNOIU