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	**Telecomunicaţii**
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicaţii şi Tehnologii Informaţionale
1.5 Ciclul de studii	Licenţă
1.6 Specializarea	Tehnologii şi Sisteme de Telecomunicaţii

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)				Programarea calculatoarelor şi limbaje de programare 2 Computer Programming and Programming Languages 2			
2.2 Titularul activităţilor de curs				Conf. Dr. Iulian Nastac			
2.3 Titularul activităţilor de seminar / laborator				Conf. Dr. Iulian Nastac			
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei		F	2.9 Codul disciplinei	04.F.02.O.011		2.10 Tipul de notare	Nota

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

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2	3.3 seminar/laborator	1
3.4 Total ore din planul de învăţământ	42	Din care: 3.5 curs	28	3.6 seminar/laborator	14
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					52
Tutorat					0
Examinări					6
Alte activităţi (dacă există):					0

3.7 Total ore studiu individual	58.00
3.8 Total ore pe semestru	100
3.9 Numărul de credite	4

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	Programarea calculatoarelor şi limbaje de programare 1
4.2 de rezultate ale învăţării	Rezolvarea unor probleme de dificultate redusă în limbajul C/C++.

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

5.1 Curs	Cursul se va desfăşura într-o sală dotată cu videoproiector şi computer.

5.2 Seminar/ Laborator/Proiect	Laborator echipat cu calculatoare si software adecvat. Prezența obligatorie la lucrările de laborator (conform regulamentului studiilor universitare de licență în UPB). Accesul individual la sisteme PC dotate cu editoare şi compilatoare C/C++ cât şi cu conexiune la Internet (acces la platforma de e-learning Moodle).

6. Obiectiv general *(Se referă la intențiile profesorilor pentru studenți, la ceea ce studenții vor fi învățaţi 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.)*

Curs: însușirea mecanismelor de abstractizare a problemelor de programare sub forma claselor şi a obiectelor. Înțelegerea modului de interacțiune între obiecte și a relațiilor dintre clase diferite. Utilizarea noțiunilor de programare obiect-orientată pentru a rezolva probleme exprimate în limbaj natural.
Laborator: însușirea practică, prin implementare de programe software, a noțiunilor predate la curs. Rezolvarea unor probleme practice concrete care includ elemente de programare obiect-orientată.

7. Competenţe *(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	- Înţelegerea şi utilizarea conceptelor fundamentale din domeniul comunicaţiilor şi transmisiunii informaţiei. - Aplicarea cunoştinţelor, conceptelor şi metodelor fundamentale privitoare la arhitectura sistemelor de calcul, microcontrolere, limbaje şi tehnici de programare. - Proiectarea şi utilizarea sistemelor de calcul şi a reţelelor de calculatoare.
Transversale (generale)	- Capacitatea de a lua decizii în vederea rezolvării problemelor curente, sau imprevizibile, care apar în procesul de exploatare a sistemelor de calcul. - Capacitatea de a se informa și documenta permanent pentru dezvoltarea personală și profesională prin citirea literaturii de specialitate. - Capacitatea de a comunica și de a prezenta conținut tehnic atât în limba română, cât și în limba engleză.

8. Rezultatele învăţării *(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.)*

Cunoştinţe	*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 faptice.* **Enumeră** principiile programării obiect-orientate. **Definește** noțiuni specifice domeniului, precum: clasă, obiect, moștenire, polimorfism. **Descrie** diferiții specificatori de acces și implicațiile acestora. **Utilizează** corect operatorii studiați și variantele lor supraîncărcate. **Interpretează şi** mplementează clase în funcție de descrieri textuale formulate în limbaj natural. **Descrie** relațiile dintre obiecte în contextul moștenirii și al polimorfismului. **Interpretează și definește** clasele în funcție de diagramele UML.

Aptitudini	*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).* **Selectează** și grupează informații referitoare la structura unei clase formulate în limbaj natural. **Utilizează argumentat principiile programării obiect-orientate.** **Foloseşte un limbaj ştiinţific adecvat.** **Verifică experimental soluții identificate.** **Rezolvă** aplicații practice. **Interpretează** adecvat relații de cauzalitate. **Analizează şi compară rezultatele obţinute cu cele dorite.** **Argumentează soluțiile identificate/modurile de rezolvare.**
Responsabilitate și autonomie	*Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale.* **Selectează surse bibliografice potrivite și le analizează.** **Respectă principiile de etică academică, citând corect sursele bibliografice utilizate.** **Demonstrează receptivitate pentru contexte noi de învățare.** **Demonstrează autonomie în organizarea situației/contextului de învățare sau a situației problemă de rezolvat.**

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.)*

Pornindu-se de la analiza caracteristicilor de învățare ale studenţilor şi de la nevoile lor specifice, procesul de predare va explora metode de predare atât expozitive (prelegerea, expunerea), cât şi conversative-interactive, bazate pe modele de învățare prin descoperire facilitate de explorarea directa şi indirectă a realității (experimentul, demonstrația, modelarea), dar şi pe metode bazate pe acțiune, precum exercițiul, activitățile practice şi rezolvarea de probleme.
În activitatea de predare vor fi utilizate prelegeri, în baza unor prezentări Power Point care vor fi puse la dispoziția studenţilor. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noțiunilor parcurse la ultimul curs.
Prezentările utilizează imagini, scheme şi exemple de cod sursă, astfel încât informațiile prezentate să fie uşor de înțeles şi asimilat.

10. Conţinuturi

CURS		
Capitolul	**Conţinutul**	**Nr. ore**
1	Introducere: aspecte generale referitoare la programarea obiect-orientată, migrarea de la limbaj de programare procedural (C) la limbaj de programare obiect-orientat (C++).	2
2	Clase & obiecte: introducerea conceptelor de clasă şi obiect. Operatori specifici C++; namespace; atribute si metode; setter & getter; constructors & destructors.	4
3	Specificatori de acces (public, private, protected). Funcții membre speciale, funcții si clase friend, rezolvare exerciții.	6
4	Moştenire: concepte de bază, ordinea apelării constructorilor/destructorilor, ascunderea membrilor, suprascrierea funcțiilor, virtualitate, ‘override’, tipuri diferite de moşteniri, rezolvare exerciții.	6

5	Casting : static cast, dinamic cast, downcasting & upcasting.	2
6	Polimorfism. Supraîncărcarea operatorilor si a funcţiilor, rezolvare exerciţii.	6
7	Verificare finală.	2
	Total:	28

Bibliografie:

D.I. Năstac, Note de curs POO, UPB, ETTI, Moodle https://curs.upb.ro/;
"The C++ Programming Language", 4th Edition, Bjarne Stroustrup
„CPP Coding Standards - 101 Rules Guidelines and Best Practices", Herb Stutter, Andrei Alexandrescu.
„Head First Object-Oriented Design and Analysis", Brett McLaughlin, Gary Pollice, David West.
„Head First Design Patterns", Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra.
C++, http://www.cplusplus.com.

LABORATOR

Nr. crt.	Conţinutul	Nr. ore
1	Clase & obiecte: introducerea conceptelor de clasă şi obiect, specificatori de acces, constructori.	2
2	Clase & obiecte: liste de iniţializare, destructori. Specificatori de acces, functii friend.	2
3	Moştenire: concepte de bază, ordinea apelării constructorilor/destructorilor, ascunderea membrilor.	2
4	Virtualitate: suprascrierea funcţiilor. Supraîncărcarea operatorilor.	2
5	Moştenire multiplă. Polimorfism. Probleme recapitulative.	2
6	Evaluarea unui miniproiect/temă de casă.	2
7	Colocviu final de laborator.	2
	Total:	14

Bibliografie:

. D.I. Năstac, Note de curs POO, UPB, ETTI, Moodle https://curs.upb.ro;
"The C++ Programming Language", 4th Edition, Bjarne Stroustrup
C++, http://www.cplusplus.com.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	Verificarea însuşirii cunoştinţelor de la curs.	Examen scris în săptămâna 14. Subiectele acoperă întreaga materie.	40%

11.5 Seminar/laborator/proiect	Rezolvarea practică cu ajutorul limbajului C++ a unor probleme de tip obiect-orientate	Colocviu final de laborator (oral şi la calculator), axat în special pe componenta practică.	30%
	Demonstrarea şi susţinerea funcţionării unui program realizat	Se evaluează un proiect (temă de casă), urmărindu-se componenta aplicativă, de deprinderi practice.	30%
11.6 Condiţii de promovare			
Participarea la lucrările de laborator; Cumularea a minim 50% din punctajul aferent disciplinei.			

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)

Programa de curs oferă absolvenţilor atât cunoştinţele necesare înţelegerii principiilor programării obiect-orientate, cât şi bazele programării folosind un limbaj general precum C++. Progresul tehnologic actual al dispozitivelor electronice şi de telecomunicaţii este condiţionat de abilitatea de dezvoltare şi experimentare folosind limbaje de programare. Astfel că disciplina de programare obiect-orientată este fundamentală în formarea viitoarelor generaţii de ingineri şi cercetători din domeniu;
Programa asigură astfel absolvenţilor competenţe adecvate cu necesităţile calificărilor actuale şi o pregătire ştiinţifică şi tehnică modernă, de calitate şi competitivă, care să le permită angajarea rapidă după absolvire. Aceasta este perfect încadrată în politica Universităţii Politehnica din Bucureşti, atât din punctul de vedere al conţinutului şi structurii, cât şi din punctul de vedere al aptitudinilor şi deschiderii internaţionale oferite studenţilor. Posibilii angajatori vizează atât mediul academic (profil didactic şi de cercetare) cât şi mediul industrial de cercetare-dezvoltare precum organizaţii/firme de orice dimensiune, de la cele mici create de studenţi/masteranzi (de exemplu start-up şi spin-off), până la cele multinaţionale.

Data completării	Titular de curs	Titular(i) de aplicaţii
21.09.2026	Conf. Dr. Iulian Nastac	Conf. Dr. Iulian Nastac

Data avizării în departament	Director de departament
22.09.2026	Conf. Dr. Serban Georgica Obreja

Data aprobării în Consiliul Facultăţii — Decan

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

22.09.2026

Prof. Dr. Mihnea Udrea

