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	**Electronică Aplicată şi Ingineria Informaţiei**
1.4 Domeniul de studii	Calculatoare şi Tehnologia Informaţiei
1.5 Ciclul de studii	Licenţă
1.6 Specializarea	Ingineria Informaţiei

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)				Programarea calculatoarelor şi limbaje de programare 2			
2.2 Titularul activităţilor de curs				S.l./Lect. Dr. Mihai DOGARIU			
2.3 Titularul activităţilor de seminar / laborator				S.l./Lect. Dr. Mihai DOGARIU			
2.4 Anul de studiu	1	2.5 Semestrul	II	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	2.00	3.3 seminar/laborator	1
3.4 Total ore din planul de învăţământ	42.00	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	Sistem de videoproiecţie pentru prezentarea cursului şi acces la platforma Moodle pentru exemplificarea rezolvării unor exerciţii.

5.2 Seminar/ Laborator/Proiect	Laborator dotat cu sisteme de calcul şi acces la platforma de e-learning Moodle. Prezența la laborator este obligatorie, conform regulament ETTI.

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.)*

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 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.)*

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.* - 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 şi documenta permanent pentru dezvoltarea personală şi profesională prin citirea literaturii de specialitate. - Capacitatea de a comunica şi de a prezenta conţinut tehnic atât în limba română, cât şi în limba engleză. Enumeră principiile programării obiect-orientate. Defineşte noţiuni specifice domeniului, precum: clasă, obiect, moştenire, polimorfism. Descrie diferiţii specificatori de acces şi implicaţiile acestora. Utilizează corect operatorii studiaţi şi variantele lor supraîncărcate. Interpretează şi mplementează clase în funcţie de descrieri textuale formulate în limbaj natural. Descrie relaţiile dintre obiecte în contextul moştenirii şi al polimorfismului. Interpretează şi defineşte clasele în funcţie de diagramele UML.
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).* Selectează şi grupează informaţii referitoare la structura unei clase formulate în limbaj natural. Utilizează argumentat principiile programării obiect-orientate. Foloseşte un limbaj ştiinţific adecvat. Verifică experimental soluţii identificate. Rezolvă aplicaţii practice. Interpretează adecvat relaţii de cauzalitate. Analizează şi compară rezultatele obţinute cu cele dorite. Argumentează soluţiile identificate/modurile de rezolvare.
Responsabilitate şi autonomie	*Capacitatea cursantului de a aplica în mod autonom şi responsabil cunoştinţele şi aptitudinile sale.* Selectează surse bibliografice potrivite şi le analizează. Respectă principiile de etică academică, citând corect sursele bibliografice utilizate. Demonstrează receptivitate pentru contexte noi de învăţare. Demonstrează autonomie în organizarea situaţiei/contextului de învăţare sau a situaţiei 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 studenţilor la stabilirea propriului parcurs de învăţare, cum se identifică eventualele rămâneri în urmă şi ce măsuri remediale se adoptă în astfel de cazuri.)*

Pornindu-se de la analiza caracteristicilor de învăţare 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 învăţare prin descoperire facilitate de explorarea directa şi indirectă a realităţii (experimentul, demonstraţia, modelarea), dar şi pe metode bazate pe acţiune, precum exerciţiul, activităţile practice şi rezolvarea de probleme.

În activitatea de predare vor fi utilizate prelegeri, în baza unor prezentări Power Point care vor fi puse la dispoziţia studenţilor. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noţiunilor parcurse la ultimul curs.

Prezentările utilizează imagini, scheme şi exemple de cod sursă, astfel încât informaţiile prezentate să fie uşor de înţeles 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++), spaţii de nume	1
2	Clase & obiecte: introducerea conceptelor de clasă şi obiect, specificatori de acces, constructori, liste de iniţializare, destructori, vectori de obiecte, funcţii membre speciale, separarea claselor în fişiere, cuvintele cheie ‘static’, ‘friend’, rezolvare exerciţii.	9
3	Moştenire: concepte de bază, ordinea apelării constructorilor/destructorilor, ascunderea membrilor, suprascrierea funcţiilor, virtualitate, cuvintele cheie ‘final’, ‘override’, tipuri diferite de moşteniri, rezolvare exerciţii.	6
4	Supraîncărcarea operatorilor: de asignare, de extragere/inserţie din/în fluxuri, incrementare/decrementare, aritmetici binari, relaţionali, rezolvare exerciţii	2
5	Templates: principiile programării generice, template de clase, template de funcţii, specializări, polimorfism.	2
6	Standard Template Library: containere (de secvenţă, asociative, asociative neordonate, adaptori de containere), iteratori, algoritmi, rezolvare exerciţii.	4
7	Lucrul cu fişiere: fluxurile de intrare/ieşire, exemple.	2
8	Verificare finală.	2
	Total:	28

Bibliografie:

Bibliografie:

1. Mihai DOGARIU, Programare Obiect-Orientată, suport de curs electronic, https://curs.upb.ro/2021/course/view.php?id=8975
2. “The C++ Programming Language”, 4th Edition, Bjarne Stroustrup
3. „CPP Coding Standards - 101 Rules Guidelines and Best Practices”, Herb Stutter, Andrei Alexandrescu.
4. „Head First Object-Oriented Design and Analysis”, Brett McLaughlin, Gary Pollice, David West.
5. „Head First Design Patterns”, Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra.

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, vectori de obiecte.	2
3	Test laborator #1	2
4	Moştenire: concepte de bază, diagramă UML, ordinea apelării constructorilor/destructorilor, ascunderea membrilor.	2
5	Virtualitate: suprascrierea funcţiilor, destructori virtuali, funcţii pur virtuale	2

6	Supraîncărcarea operatorilor, Standard Template Library	2
7	Test laborator #2	2
	Total:	14

Bibliografie:
Mihai DOGARIU, Programare Obiect-Orientată, suport de laborator electronic, https://curs.upb.ro/2021/course/view.php?id=8975

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	Capacitatea de a rezolva o problemă de programare în limbajul C/C++ folosind noţiuni de programare obiect-orientată.	Examen final, susţinut practic pe platforma Moodle. Studentul rezolvă cerinţele propuse şi este analizat modul de rezolvare al acestora şi soluţiile propuse (sesiune).	50%
11.5 Seminar/laborator/proiect	Capacitatea de a rezolva o problemă de programare în limbajul C/C++ folosind noţiuni de programare obiect-orientată.	Lucrare intermediară de laborator, susţinută practic pe platforma Moodle. Studentul rezolvă cerinţele propuse şi este analizat modul de rezolvare al acestora şi soluţiile propuse (săptămânile 5&6).	20%
	Capacitatea de a rezolva o problemă de programare în limbajul C/C++ folosind noţiuni de programare obiect-orientată.	Lucrare finală de laborator, susţinută practic pe platforma Moodle. Studentul rezolvă cerinţele propuse şi este analizat modul de rezolvare al acestora şi soluţiile propuse (săptămânile 13&14).	30%
11.6 Condiţii de promovare			
- Participarea la lucrările de laborator. - Cumularea a minim 50% din punctajul aferent laboratorului. - 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 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	S.l./Lect. Dr. Mihai DOGARIU	S.l./Lect. Dr. Mihai DOGARIU

Data avizării în departament	Director de departament

Data aprobării în Consiliul Facultăţii	Decan
22.09.2026	Prof. Dr. Mihnea Udrea