



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie Politehnica București
1.2 Facultatea	Electronică, Telecomunicații și Tehnologia Informației
1.3 Departamentul	Electronică Aplicată și Ingineria Informației
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Specializarea	Ingineria Informației

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)	Programarea calculatoarelor și limbaje de programare 2						
2.2 Titularul activităților de curs	S.l./Lect. Dr. Mihai DOGARIU						
2.3 Titularul activităților 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 activităților 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 învățământ	42.00	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					52
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					
Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					
Tutorat					0
Examinări					6
Alte activități (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. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea Calculatoarelor și Limbaje de Programare 1
4.2 de rezultate ale învățării	Rezolvarea unor probleme de dificultate redusă în limbajul C/C++.

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

5.1 Curs	Sistem de videoproiecție pentru prezentarea cursului și acces la platforma Moodle pentru exemplificarea rezolvării unor exerciții.
----------	--



5.2 Seminar/ Laborator/Proiect	Laborator dotat cu sisteme de calcul și 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 și 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. Competențe (*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	<ul style="list-style-type: none">- Înțelegerea și utilizarea conceptelor fundamentale din domeniul comunicațiilor și transmisiunii informației.- Aplicarea cunoștințelor, conceptelor și metodelor fundamentale privitoare la arhitectura sistemelor de calcul, microcontrolere, limbaje și tehnici de programare.- Proiectarea și utilizarea sistemelor de calcul și a rețelelor de calculatoare.
Transversale (generale)	<ul style="list-style-type: none">- 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 exemplurilor 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.)*



Cunoștințe	<p><i>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.</i></p> <ul style="list-style-type: none">- 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. <p>Enumeră principiile programării obiect-orientate. Definește noțiuni specifice domeniului, precum: clasă, obiect, moștenire, polimorfism. Describe diferenții specificatori de acces și implicațiile acestora. Utilizează corect operatorii studiați și variantele lor supraîncărcate. Interpretează și implementează clase în funcție de descrieri textuale formulate în limbaj natural. Describe relațiile dintre obiecte în contextul moștenirii și al polimorfismului. Interpretează și definește clasele în funcție de diagramele UML.</p>
Aptitudini	<p><i>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).</i></p> <p>Selectează și grupează informații referitoare la structura unei clase formulate în limbaj natural. Utilizează argumentat principiile programării obiect-orientate. Folosește un limbaj științific adecvat. Verifică experimental soluții identificate. Rezolvă aplicații practice. Interpretează adecvat relații de cauzalitate. Analizează și compară rezultatele obținute cu cele dorite. Argumentează soluțiile identificate/modurile de rezolvare.</p>
Responsabilitate și autonomie	<p><i>Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale.</i></p> <p>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.</p>

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 rămăneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)

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.



Pornindu-se de la analiza caracteristicilor de învățare ale studenților și de la nevoile lor specifice, procesul de predare va explora metode de predare atât expositive (prelegerea, expunerea), cât și conversative-interactive, bazate pe modele de învățare prin descoperire facilitate de explorarea directă și indirectă a realității (experimentul, demonstrația, modelarea), dar și pe metode bazate pe acțiune, precum exercițiul, activitățile practice și rezolvarea de probleme.

În activitatea de predare vor fi utilizate prelegeri, în baza unor prezentări Power Point 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 imagini, scheme și exemple de cod sursă, astfel încât informațiile prezentate să fie ușor de înțeles și asimilat.

10. Conținuturi

CURS		
Capitolul	Conținutul	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++), spații de nume	1
2	Clase & obiecte: introducerea conceptelor de clasă și obiect, specificatori de acces, constructori, liste de inițializare, destructori, vectori de obiecte, funcții membre speciale, separarea claselor în fișiere, cuvintele cheie 'static', 'friend', rezolvare exerciții.	9
3	Moștenire: concepte de bază, ordinea apelării constructorilor/destructorilor, ascunderea membrilor, suprascrierea funcțiilor, virtualitate, cuvintele cheie 'final', 'override', tipuri diferite de moșteniri, rezolvare exerciții.	6
4	Supraîncărcarea operatorilor: de asignare, de extragere/insertie din/în fluxuri, incrementare/decrementare, aritmetici binari, relaționali, rezolvare exerciții	2
5	Templates: principiile programării generice, template de clase, template de funcții, specializări, polimorfism.	2
6	Standard Template Library: containere (de secvență, asociative, asociative neordonate, adaptor de containere), iteratori, algoritmi, rezolvare exerciții.	4
7	Lucrul cu fișiere: fluxurile de intrare/ieșire, 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.	Conținutul	Nr. ore
1	Clase & obiecte: introducerea conceptelor de clasă și obiect, specificatori de acces, constructori.	2
2	Clase & obiecte: liste de inițializare, destructori, vectori de obiecte.	2
3	Test laborator #1	2
4	Moștenire: concepte de bază, diagramă UML, ordinea apelării constructorilor/destructorilor, ascunderea membrilor.	2
5	Virtualitate: suprascrierea funcțiilor, destructori virtuali, funcții 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 noțiuni de programare obiect-orientată.	Examen final, susținut practic pe platforma Moodle. Studentul rezolvă cerințele propuse și este analizat modul de rezolvare al acestora și soluțiile propuse (sesiune).	50%
11.5 Seminar/laborator/proiect	Capacitatea de a rezolva o problemă de programare în limbajul C/C++ folosind noțiuni de programare obiect-orientată.	Lucrare intermediară de laborator, susținută practic pe platforma Moodle. Studentul rezolvă cerințele propuse și este analizat modul de rezolvare al acestora și soluțiile propuse (săptămânile 5&6).	20%
	Capacitatea de a rezolva o problemă de programare în limbajul C/C++ folosind noțiuni de programare obiect-orientată.	Lucrare finală de laborator, susținută practic pe platforma Moodle. Studentul rezolvă cerințele propuse și este analizat modul de rezolvare al acestora și soluțiile propuse (săptămânile 13&14).	30%
11.6 Condiții 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 conținutului disciplinei cu așteptările reprezentanților angajatorilor și asociațiilor profesionale reprezentative din domeniul aferent programului, precum și cu stadiul actual al cunoașterii în domeniul științific abordat și practicile în instituții de învățământ superior din Spațiul European al Învățământului Superior (SEİS)

Programa asigură astfel absolvenților competențe adecvate cu necesitățile calificărilor actuale și o pregătire științifică și tehnică modernă, de calitate și competitivă, care să le permită angajarea rapidă după absolvire. Aceasta este perfect încadrată în politica Universității Politehnica din București, atât din punctul de vedere al conținutului și structurii, cât și din punctul de vedere al aptitudinilor și deschiderii internaționale oferite studenților. Posibili angajatori vizează atât mediul academic (profil didactic și de cercetare) cât și mediul industrial de cercetare-dezvoltare precum organizații/firme de orice dimensiune, de la cele mici create de studenți/masteranzi (de exemplu start-up și spin-off), până la cele multinaționale.

Data completării	Titular de curs	Titular(i) de aplicații
14.10.2024	S.I./Lect. Dr. Mihai DOGARIU 	S.I./Lect. Dr. Mihai DOGARIU 
Data avizării în departament	Director de departament	
16.10.2024	Conf. Dr. Bogdan Cristian FLOREA 	
Data aprobării în Consiliul Facultății	Decan	
01.11.2024	Prof. Dr. Mihnea Udrea 