



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



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	Dispozitive, Circuite și Arhitecturi Electronice
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
1.5 Ciclu de studii	Licență
1.6 Specializarea	Electronică aplicată

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)				Programarea calculatoarelor și limbaje de programare 2 Computer Programming and Programming Languages 2			
2.2 Titularul activităților de curs				Conf. Dr. Radu Hobincu			
2.3 Titularul activităților de seminar / laborator				Conf. Dr. Radu Hobincu			
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 Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					56
Tutorat					2
Examinări					0
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	Gândire logică, structurată. Cunoștințe de sintaxă a limbajului C.

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

5.1 Curs	Sală de curs cu proiector și acces Internet
----------	---



5.2 Seminar/ Laborator/Proiect	Laborator cu calculatoare ce rulează un sistem de operare Linux, ce dispune minim de compilator de C++ (g++) și editor de text, de preferat un IDE modern (ex: JetBrains CLion). Capacitatea laboratorului trebuie să fie de minim 15 locuri și să aibă acces la Internet.
-----------------------------------	--

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.)*

Obiectivul general al cursului este studiul unei noi paradigme de programare: orientarea pe obiecte precum și a modurilor prin care se pot dezvolta aplicații software utilizând aceste modele, prin implementarea cu ajutorul claselor și obiectelor. Astfel, se urmărește acumularea de cunoștințe în domeniul arhitecturii software orientate pe obiecte presupunând și asimilarea unui nou limbaj de programare: C++. În plus, exemplele și temele propuse sunt utile pentru asimilarea API-ului C++ care la rândul lui permite dezvoltarea rapidă și eficientă a soluțiilor software performante.

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	C3. Aplicarea cunoștințelor, conceptelor și metodelor de bază privitoare la arhitectura sistemelor de calcul, microprocesoare, microcontrolere, limbaje și tehnici de programare.
Transversale (generale)	CT1 Analiza metodică a problemelor întâlnite în activitate, identificând elementele pentru care există soluții consacrate, asigurând astfel îndeplinirea sarcinilor profesionale. CT3 Adaptarea la noile tehnologii, dezvoltarea profesională și personală, prin formare continuă folosind surse de documentare tipărite, software specializat și resurse electronice în limba română și, cel puțin, într-o limbă de circulație internațională.

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.)*

Cunoștințe	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. Cunoștințe legate de sintaxa limbajului C++ și de implementarea aplicațiilor structurate pe clase.
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). Aptitudini legate de a analiza o problemă dată în limbaj natural și a dezvolta o soluție proprie pentru rezolvarea problemei, precum și capacitatea de a implementa această soluție în limbajul C++, structurând codul conform paradigmei programării orientate pe obiecte.



Responsabilitate
și autonomie

Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale. Capacitatea de a căuta răspunsuri și a se documenta online legat de sintaxa și API-ul limbajului C++, precum și de a căuta soluții la probleme punctuale.

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

După prezentarea teoretică a noțiunilor necesare, cu ajutorul proiecteurului și a unui calculator, se exemplifică practic prin scrierea de mici programe și analiza rezultatelor. Se cere constant feedback studenților în legătură cu calitatea explicațiilor și pentru sugestii de implementare a exemplelor. Noțiunile teoretice sunt prezentate cu ajutorul unei tablete și Stylus care proiectează informația pe ecran, urmând ca la sfârșitul cursului, notițele să fie publicate pe Moodle. În laborator studenții beneficiază de un calculator și un set de exerciții pe care le rezolvă individual cu ajutorul cadrului didactic și a platformelor de laborator.

10. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore
1	Recapitulare pointeri și structuri	2
2	Introducere în programarea orientată pe obiecte - clasa, metoda, câmpul și obiectul	2
3	Constructorul, supraîncărcarea și încapsularea (public/ private)	2
4	Destructorul, supraîncărcarea operatorilor și referința	4
5	Metode const-qualified, constructorul de copiere, operatorul de copiere prin atribuire, regula de 3	2
6	Șabloane de clase (templates) și spații de nume (namespaces)	4
7	Moștenirea și modificatorul de acces protected. Metode virtuale, pur virtuale și clase abstracte	4
8	Tratarea excepțiilor	2
9	Expresii regulate	2
10	Introducere în programare paralelă (std::thread. std::future). Bariera (std::mutex)	2
11	Semaforul și variabila atomică	2
Total:		28

Bibliografie:

1. Radu HOBINCU, Călin BÎRĂ, Programare Orientată pe Obiecte, suport de curs electronic, <https://curs.upb.ro/>
2. Stanley B. Lippman , Josee Lajoie, Barbara E. Moo, " C++ Primer, 5th Edition", Addison Wesley, ISBN: 9780321714114, 2012
3. <https://en.cppreference.com/w/cpp> - Descrierea API-ului limbajului C++



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



LABORATOR		
Nr. crt.	Conținutul	Nr. ore
1	Recapitulare – utilizarea debugger-ului în exerciții cu pointeri și structuri	2
2	Exerciții cu clase, obiecte, metode, câmpuri și constructori	2
3	Exerciții cu supraîncărcare de operatori, destructori și referințe	2
4	Exerciții cu moștenire, polimorfism și metode (pur) virtuale	2
5	Exerciții cu excepții	2
6	Simulare de colocviu de laborator	2
7	Colocviu de laborator	2
	Total:	14
Bibliografie: Platformă de laborator - https://wiki.dcae.pub.ro/index.php/Programare_Orientat%C4%83_pe_Obiecte_(C%2B%2B)		

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
11.4 Curs	Însușirea noțiunilor teoretice și de sintaxă C++ pe parcursul semestrului.	Patru teme de casă, a câte cinci puncte fiecare, de complexitate medie spre ridicată, a căror soluții se testează automat pe platforma Moodle.	10
	Însușirea noțiunilor teoretice în ansamblu.	Test grilă, susținut pe platforma Moodle.	40
11.5 Seminar/laborator/proiect	Capacitatea studentului de a folosi noțiunile acumulate pentru a rezolva probleme practice.	Rezolvarea de probleme de programare în timpul fiecărui laborator.	20
	Capacitatea de a abstractiza o problemă reală și a implementa o soluție în limbajul C++.	Colocviu de laborator, susținut pe platforma Moodle, în care se cere rezolvarea individual a unei probleme simple, similare cu cele întâlnite în laborator, cu cerința specifică ca aceasta să fie implementată corect și complet pentru acordarea punctajului.	30
11.6 Condiții de promovare			
- Obținerea a 50% din punctajul total. - Obținerea a 50% din punctajul aferent activității de laborator.			



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)

Programarea obiect-orientată este unul din cei mai importanți pași făcuți în evoluția limbajelor de programare spre o mai puternică abstractizare în implementarea programelor. În acest context, disciplina Programare Obiect-Orientată întregeste linia disciplinelor de programare, începută cu Programarea Calculatoarelor și finalizată cu Structuri de Date și Algoritmi, și reprezintă baza pentru a înțelege și utiliza abordările mai noi din programare: cea bazată pe componente și cea bazată pe servicii. Programa cursului răspunde concret acestor cerințe actuale de dezvoltare și evoluție, subscrise economiei europene a serviciilor din domeniul Calculatoare și Tehnologia Informației (CTI). Se asigură astfel absolvenților competențe adecvate cu necesitățile calificărilor actuale și o pregătire științifică și tehnică moderne, de calitate și competitive, care să le permită angajarea rapidă după absolvire, fiind perfect încadrat în politica Universității Naționale de Știință și Tehnologie Politehnica București, atât din punctul de vedere al conținutului și structurii, cât și din punctul de vedere al aptitudinilor și deschiderii internaționale oferite studenților.

Data completării

Titular de curs

Titular(i) de aplicații

25.09.2025

Conf. Dr. Radu Hobincu

As. Drd. Ing. Alexandra ENESCU

Data avizării în departament

Director de departament

Prof. Dr. Ing. Claudius DAN

Data aprobării în Consiliul Facultății

Decan

26.09.2025

Prof. Dr. Mihnea Udrea