**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 | **Dispozitive, Circuite şi Arhitecturi Electronice** |
| 1.4 Domeniul de studii | Inginerie Electronică, Telecomunicaţii şi Tehnologii Informaţionale |
| 1.5 Ciclul de studii | Licenţă |
| 1.6 Specializarea | Electronică aplicată |

## 2. Date despre disciplină

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

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

| | |
|---|---|
| 5.1 Curs | Sală de curs cu proiector şi 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++) şi editor de text, de preferat un IDE modern (ex: Jetbrains CLion). Capacitatea laboratorului trebuie să fie de minim 15 locuri şi să aibă acces la Internet. |

**6. Obiectiv general** *(Se referă la intenţiile profesorilor pentru studenţi, la ceea ce studenţii vor fi învăţaţi în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului ştiinţific abordat, precum şi 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 învăţământ al specializării studiate etc.)*

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

**7. Competenţe** *(Capacitatea dovedită de a utiliza cunoştinţe, aptitudini şi abilităţi personale, sociale şi/sau metodologice în situaţii de muncă sau de studiu şi pentru dezvoltarea profesională şi personală. Reflectă cerinţele angajatorilor.)*

| | |
|---|---|
| **Specifice** | C3. Aplicarea cunoştinţelor, conceptelor şi metodelor de bază privitoare la arhitectura sistemelor de calcul, microprocesoare, microcontrolere, limbaje şi tehnici de programare. |
| **Transversale (generale)** | CT1 Analiza metodică a problemelor întâlnite în activitate, identificând elementele pentru care există soluţii consacrate, asigurând astfel îndeplinirea sarcinilor profesionale.<br>CT3 Adaptarea la noile tehnologii , dezvoltarea profesionala şi personală, prin formare continuă folosind surse de documentare tipărite, software specializat şi resurse electronice în limba română şi, cel puţin, într-o limbă de circulaţie internaţională. |

**8. Rezultatele învăţării** *(Sunt enunţuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învăţării reflectă realizările studentului şi mai puţin intenţiile profesorului. Rezultatele învăţării informează studenţii despre ceea ce se aşteaptă de la ei din punct de vedere al performanţei, pentru a obţine notele şi creditele dorite. Sunt definite în termeni concreţi, folosind verbe similare exemplelor de mai jos şi indică ceea ce se va urmări prin evaluare. Rezultatele învăţării vor fi astfel redactate încât să fie evidenţiată clar relaţia faţă de competenţele 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.*<br>Cunoştinţe legate de sintaxa limbajului C++ şi de implementarea aplicaţiilor structurate pe clase. |
| **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).*<br>Aptitudini legate de a analiza o problem dată în limbaj natural şi a dezvolta o soluţie proprie pentru rezolvarea problemei, precum şi capacitatea de a implementa această soluţie 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 şi a se documenta online legat de sintaxa şi API-ul limbajului C++, precum şi de a căuta soluţii 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 rămâneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)*

După prezentarea teoretică a noţiunilor necesare, cu ajutorul proiectorului şi a unui calculator, se exemplifică practic prin scrierea de mici programe şi analiza rezultatelor. Se cere constant feedback studenţilor în legătură cu calitatea explicaţiilor şi pentru sugestii de implementare a exemplelor. Noţiunile teoretice sunt prezentate cu ajutorul unei tablete şi Stylus care proiectează informaţia pe ecran, urmând ca la sfârşitul cursului, notiţele să fie publicate pe Moodle. În laborator studenţii beneficiază de un calculator şi un set de exerciţii pe care le rezolvă individual cu ajutorul cadrului didactic şi a platformelor de laborator.

**10. Conţinuturi**

| **CURS** | | |
|---|---|---|
| **Capitolul** | **Conţinutul** | **Nr. ore** |
| 1 | Recapitulare pointeri şi structuri | 2 |
| 2 | Introducere în programarea orientată pe obiecte - clasa, metoda, câmpul şi obiectul | 2 |
| 3 | Constructorul, supraîncărcarea si incapsularea (public/ private) | 2 |
| 4 | Destructorul, supraîncărcarea operatorilor şi referinţa | 4 |
| 5 | Metode const-qualified, constructorul de copiere, operatorul de copiere prin atribuire, regula de 3 | 2 |
| 6 | Şabloane de clase (templates) şi spaţii de nume (namespaces) | 4 |
| 7 | Moştenirea şi modifcatorul de acces protected. Metode virtuale, pur virtuale şi clase abstracte | 4 |
| 8 | Tratarea excepţiilor | 2 |
| 9 | Expresii regulate | 2 |
| 10 | Introducere în programare paralelă (std::thread. std::future). Bariera (std::mutex) | 2 |
| 11 | Semaforul şi 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+

| **LABORATOR** | | |
|---|---|---|
| **Nr. crt.** | **Conţinutul** | **Nr. ore** |
| 1 | Recapitulare – utilizarea debugger-ului în exerciţii cu pointeri şi structuri | 2 |
| 2 | Exerciţii cu clase, obiecte, metode, câmpuri şi constructori | 2 |
| 3 | Exerciţii cu supraîncărcare de operatori, destructori şi referinţe | 2 |
| 4 | Exerciţii cu moştenire, polimorfism şi metode (pur) virtuale | 2 |
| 5 | Exerciţii cu excepţii | 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 | Însuşirea noţiunilor teoretice şi de sintaxă C++ pe parcursul semestrului. | Patru teme de casă, a câte cinci puncte fiecare, de complexitate medie spre ridicată, a căror soluţii se testează automat pe platforma Moodle. | 10 |
| | Însuşirea noţiunilor teoretice în ansamblu. | Test grilă, susţinut pe platforma Moodle. | 40 |
| 11.5 Seminar/laborator/proiect | Capacitatea studentului de a folosi noţiunile acumulate pentru a rezolva probleme practice. | Rezolvarea de probleme de programare în timpul fiecărui laborator. | 20 |
| | Capacitatea de a abstractiza o problemă reală şi a implementa o soluţie în limbajul C++. | Colocviu de laborator, susţinut pe platforma Moodle, în care se cere rezolvarea individual a unei probleme simple, similare cu cele întâlnite în laborator, cu cerinţa specifică ca aceasta să fie implementată corect şi complet pentru acordarea punctajului. | 30 |
| 11.6 Condiţii de promovare | | | |
| • Obţinerea a 50% din punctajul total.<br>• Obţinerea a 50% din punctajul aferent activităţii de laborator. | | | |

**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)**

Programarea obiect-orientată este unul din cei mai importanţi paşi făcuţi în evoluţia limbajelor de programare spre o mai puternică abstractizare în implementarea programelor. În acest context, disciplina Programare Obiect-Orientată întregeşte linia disciplinelor de programare, începută cu Programarea Calculatoarelor şi finalizată cu Structuri de Date şi Algoritmi, şi reprezintă baza pentru a înţelege şi utiliza abordările mai noi din programare: cea bazată pe componente şi cea bazată pe servicii. Programa cursului răspunde concret acestor cerinţe actuale de dezvoltare şi evoluţie, subscrise economiei europene a serviciilor din domeniul Calculatoare şi Tehnologia Informaţiei (CTI). Se asigură astfel absolvenţilor competenţe adecvate cu necesităţile calificărilor actuale si o pregătire ştiinţifică şi tehnică moderne, de calitate şi competitive, care să le permită angajarea rapidă după absolvire, fiind perfect încadrat în politica Universităţii Naţionale de Ştiinţă şi Tehnologie Politehnica Bucureşti, atât din punctul de vedere al conţinutului şi structurii, cât şi din punctul de vedere al aptitudinilor şi deschiderii internaţionale oferite studenţilor.

| Data completării | Titular de curs | Titular(i) de aplicaţii |
|---|---|---|
| 21.09.2026 | 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 Facultăţii | Decan |
|---|---|
| 22.09.2026 | Prof. Dr. Mihnea Udrea |