

FIȘA DISCIPLINEI

Programare orientată obiect

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare orientată obiect				Codul disciplinei	MLR5006
2.2. Titularul activităților de curs		Prof. dr. CZIBULA Istvan Gergely					
2.3. Titularul activităților de seminar		Prof. dr. CZIBULA Istvan Gergely					
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligativu

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	1 sem 2 lab
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					18
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					8
Tutoriat (consiliere profesională)					7
Examinări					10
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				55	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Fundamentele programării, Structuri de date și algoritmi
4.2. de competențe	Cunoștințe medii de programare într-un limbaj de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	Sală de laborator cu calculatoare dotate cu limbajul de programare C++ și biblioteca QT

6.1. Competențele specifice acumulate¹

Competențe profesionale/ esențiale	• utilizarea bazelor teoretice ale informaticii și a modelelor formale • utilizarea instrumentelor informatice în context interdisciplinar • programarea în limbaje de nivel înalt
Competențe transversale	• aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: • metodele, algoritmi, paradigme și tehnicile folosite în diferite ramuri ale informaticii. • utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor.
Aptitudini	Studentul este capabil să: • prezinte și să explice metodele, algoritmi, paradigme și tehnicile folosite în diferite ramuri ale informaticii. • folosească paradigme de programare (procedural, orientat obiect, funcțional) pentru realizarea de aplicații software adecvate specificului domeniului aplicației dezvoltate. • înțeleagă și să comunice eficient informațiile.
Responsabilități și autonomie	Studentul are capacitatea de a lucra independent pentru: • a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu. • a înțelege și folosi conceptelor programării orientate obiect la dezvoltarea unor aplicații software de complexitate medie. • conceperea programelor de calculator și analiza sistemelor software.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Să deprindă studentul cu proiectare orientată obiect a problemelor de scară mică/mijlocie și învățarea limbajului de programare C++ și crearea de interfețe grafice utilizator în QT.
---------------------------------------	---

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

7.2 Obiectivele specifice	• Demonstrarea diferenței dintre proiectarea imperativă tradițională și proiectarea orientată obiect. • Explicarea structurilor de tip clasă ca fiind componente fundamentale, modulare. • Înțelegerea rolului moștenirii, polimorfismului, legării dinamice și a structurilor generice în realizarea codului reutilizabil. • Explicarea și utilizarea dezvoltării bazate pe funcționalități, dezvoltarea bazată pe testare, utilizarea aserțiunilor formale și tratarea excepțiilor. • Scrierea de programe de scară mică/mijlocie folosind C++ și QT. • Utilizarea claselor scrise de alți programatori în dezvoltarea sistemelor.
----------------------------------	---

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Elemente de bază ale limbajului C. • Elemente de bază ale limbajului C. • Elemente lexicale. Operatori. Conversii. • Tipuri de date. Variabile. Constante. • Domeniul de vizibilitate și durata de viață a variabilelor. Spații de nume. • Declararea și definirea funcțiilor. Supraîncărcarea funcțiilor. Funcții inline.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
2. Programare modulară în C/C++. • Funcții. Parametri. • Fișiere header. Biblioteci. • Implementarea modulară a TAD-urilor. • Utilizarea pointerului void pentru obținerea genericității.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
3. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++. • Tipuri de date: vectori și structuri. • Tipuri de date: pointeri și referințe. • Gestiunea memoriei în C/C++ • Pointeri la funcții și pointeri spre void.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
4. Programare orientată obiect în C++. • Clase și obiecte. • Membrii unei clase. Modificatori de acces. • Constructori/destructori. • Gestiunea memoriei în C++ (RAII) • Implementarea TAD-urilor în C++ • Diagrame UML pentru clase (membri, acces).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
5. Elemente de programare generică • Funcții/clase parametrizate. Mecanismul de template din C++ • Implementare TAD-uri folosind clase parametrizate • Containere și iteratori – biblioteca STL	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
6 Moștenire • Moștenire simplă. Clase derivate. • Principiul substituției. • Supraîncărcarea metodelor. • Moștenire multiplă. • Relații de specializare/generalizare – reprezentări UML.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
7 Polimorfism • Moștenire, polimorfism • Stream-uri I/O. Ierarhia de clase I/O. • Formatare. Manipulatori.	• Expunerea interactivă • Explicația • Conversația • Demonstrația	

Fișiere text.	didactică	
8 Ierarhii de clase - Excepții, spații de nume - Fișiere text.	- Expunerea interactivă - Explicația - Conversația - Demonstrația didactică	
9 Interfețe grafice utilizator - QT Toolkit: instalare, instrumente și module Qt - Componente grafice utilizator - Layout management	- Expunerea interactivă - Explicația - Conversația - Demonstrația didactică	
10 Elemente de programare bazată pe evenimente - Evenimente: Semnale și sloturi Qt - Proiectare GUI - Callback/Observer	- Expunerea interactivă - Explicația - Conversația - Demonstrația didactică	
11 Elemente de programare bazată pe evenimente - Componente grafice cu modele - Șablonul MVC. - Studiu de caz. Detalii comenzi – Produse.	- Expunerea interactivă - Explicația - Conversația - Demonstrația didactică	
12 Șabloane de proiectare. - Șabloane de proiectare creaționale, structurale, comportamentale. - Exemple	- Expunerea interactivă - Explicația - Conversația - Demonstrația didactică	
13. Aplicație POS (Point Of Sale) - Șabloane de proiectare Façade, Strategy. - Șablonul de proiectare Composite.	- Expunerea interactivă - Explicația - Conversația - Demonstrația didactică	
14. Recapitulare	- Expunerea interactivă - Conversația	

Bibliografie

1. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998.
2. Bruce Eckel, Thinking in C++, www.bruceeckel.com
3. Alexandrescu, Programarea moderna în C++. Programare generica și modele de proiectare aplicate, Editura Teora, 2002
4. M. Frentiu, B. Parv, Elaborarea programelor. Metode și tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994.
5. E. Horowitz, S. Sahni, D. Mehta, Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995.
6. K.A. Lambert, D.W. Nance, T.L. Naps, Introduction to Computer Science with C++, West Publishing Co., New-York, 1996.
7. L. Negrescu, Limbajul C++, Ed. Alabastra, Cluj-Napoca 1996.

8.2 Seminar	Metode de predare	Observații
		Seminarul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
1. Probleme simple în C. Funcții și parametri. Variabile locale și globale. Vizibilitate. Vectori (uni și multidimensionali) și structuri.	- Expunere interactivă - Explicatie - Conversatie - Exemple - Demonstratie didactica	
2. TAD de tip container cu elemente generice (void*): reprezentări vizibile și ascunse.	- Expunere interactivă - Explicatie - Conversatie - Exemple - Demonstratie didactica	

3. Clase. Clase simple. Supraîncărcarea operatorilor. Clase cu obiecte ca date membre.	• Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica	
4. Clase de tip vector dinamic și iteratori. Moștenire. Polimorfism	• Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica	
5. Containere , iteratori, algoritmi STL	• Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica	
6. Interfete grafice utilizator	• Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica	
7. Probleme complexe implementate pe baza diagramelor UML. Șabloane de proiectare. Pregătire pentru examenul scris.	• Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica	
8.3 Laborator	Metode de predare	Observații
	•	• Laboratorul este structurat sub forma a 2 ore săptămânal. • Documentațiile de laborator și programele se predau în săptămâna următoare celei în care s-a dat tema
1. Instalare MinGW și Eclipse CDT. Visual Studio. Specificare, proiectare și implementare probleme simple în C/C++. Aspecte generale C/C++.	• Lucrare de laborator • Explicația • Conversația • Modelarea	
2. Programare simple în C	• Lucrare de laborator • Explicația • Conversația • Modelarea	
3. Programare modulara C/C++	• Lucrare de laborator • Explicația • Conversația • Modelarea	
4. Proces de dezvoltare “feature driven” , arhitectura stratificata, test driven development	• Lucrare de laborator • Explicația • Conversația • Modelarea	
5. Proces de dezvoltare “feature driven” , arhitectura stratificata, test driven development	• Lucrare de laborator • Explicația • Conversația • Modelarea	
6. Programare orientata obiect	• Lucrare de laborator • Explicația	

	• Conversația • Modelarea	
7. Programare orientată obiect	• Lucrare de laborator • Explicația • Conversația • Modelarea	
8. Containere, iteratori și algoritmi STL	• Lucrare de laborator • Explicația • Conversația • Modelarea	
9. Fișiere text	• Lucrare de laborator • Explicația • Conversația • Modelarea	
10. GUI folosind QT	• Lucrare de laborator • Explicația • Conversația • Modelarea	
11. GUI folosind QT	• Lucrare de laborator • Explicația • Conversația • Modelarea	
12 Componente cu modele, MVC, Observer	• Lucrare de laborator • Explicația • Conversația • Modelarea	
13. Predare laboratoare (a se vedea observația)	• Lucrare de laborator • Explicația • Conversația • Modelarea	
14. Predare laboratoare (a se vedea observația)	• Lucrare de laborator • Explicația • Conversația • Modelarea	

Bibliografie

1. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998.
2. Bruce Eckel, Thinking in C++, www.bruceeckel.com
3. Alexandrescu, Programarea modernă în C++. Programare generică și modele de proiectare aplicate, Editura Teora, 2002
4. M. Frentiu, B. Parv, Elaborarea programelor. Metode și tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994.
5. E. Horowitz, S. Sahni, D. Mehta, Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995.
6. K.A. Lambert, D.W. Nance, T.L. Naps, Introduction to Computer Science with C++, West Publishing Co., New-York, 1996.
7. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca 1996

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Conținutul disciplinei este în concordanță cu recomandările ACM pentru domeniul informatică.
- Cursul există în programul de studiu al universităților importante din țară și străinătate.
- Conținutul disciplinei este considerat de companiile software ca important pentru a asigura cunoștințe medii de programare orientată obiect.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	Corectitudinea și completitudinea cunoștințelor acumulate și capacitatea de a proiecta și implementa programe C++	Examen scris (în sesiune)	40%
10.5 Seminar/laborator	Capacitatea de a proiecta, testa și depana programe C++ folosind QT	Evaluare practică (în sesiune)	30%
	Corectitudinea programelor C++ și a documentațiilor de laborator	-documentații -portofoliu -observare continuă	30%
10.7 Standard minim de performanță			
- Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaștere și înțelegere a domeniului, că este capabil să exprime cunoștințele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni și de a utiliza cunoștințele în rezolvarea unor probleme în limbajul de programare C++. - Pentru promovare, este obligatorie prezența la minimum 5 seminarii și 12 laboratoare . - Pentru promovare este necesar ca notele pe toate activitățile sa fie cel puțin 5 și nota finală să fie cel puțin 5.			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

10.04.2025

Semnătura titularului de curs

Prof. dr. CZIBULA Istvan Gergely

Semnătura titularului de seminar

Prof. dr. CZIBULA Istvan Gergely

Data avizării în departament:

Semnătura directorului de departament

Conf. dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică*".