

FIȘA DISCIPLINEI

Programare orientată obiect

Anul universitar 2026-2027

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	Examen
2.7. Regimul disciplinei	Obligativu		2.8. Tipul disciplinei	Disciplină fundamentală (DF)	

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 dobândite în urma absolvirii programului de studii (se preiau din planul de învățământ)¹

Competențe profesionale	
Codul competenței	Competență
CP1	Creează softuri
CP4	Definește arhitectura software
CP5	Definește cerințe tehnice
Competențe transversale	
Codul competenței	Competență
CT2	Soluționează probleme
CT3	Gândește analitic

6.2. Rezultatele învățării specifice programului de studii (se preiau din planul de învățământ)²

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP1	1. Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.	1. Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
CP4	2. Studentul/absolventul cunoaște și explică paradigmele moderne de programare, arhitecturi software și metodologii de dezvoltarea proiectelor software.	2. Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software scalabile și utilizează eficient resursele hardware și software.
CP5	3. Studentul/absolventul alege, explică și specifică fundamentele matematice aplicate în informatică, inclusiv logica formală, algebra, probabilitățile și statisticile.	3. Studentul/absolventul aplică, evaluează, propune metodele matematice pentru modelarea, simularea și rezolvarea problemelor informatice.
CT2	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții.
CT3	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)

¹ Se vor prelua din Planul de învățământ al programului de studii acele competențe profesionale și/sau transversale la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa disciplinei. Pentru fiecare competență se va prelua întregul enunț, inclusiv codul competenței, cu formularea care apare în planul de învățământ, fără modificări. Dacă nu se preia nici o competență din oricare din cele două categorii, se șterge linia din tabel aferentă acelei categorii.

² Se menționează rezultatele învățării specifice programului de studiu la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa. Enunțurile, preluate fără modificări din Planul de învățământ în funcție de tipul disciplinei (DF/DS/DC) se trec în dreptul competenței asociate.

1. Studentul cunoaște conceptele de bază ale proiectării orientate pe obiecte a problemelor de scară mică/mijlocie.
2. Studentul cunoaște limbajul de programare C++ și crearea de interfețe grafice utilizator în QT.
2. Studentul înțelege diferența dintre proiectarea imperativă tradițională și proiectarea orientată pe obiecte.
Abilități academice specifice (Specific academic skills)
1. Studentul este capabil să folosească structurile de tip clasă ca fiind componente fundamentale, modulare.
2. Studentul este capabil să scrie programe de scară mică/mijlocie folosind C++ și QT.
3. Studentul este capabil să utilizeze dezvoltarea bazată pe funcționalități, dezvoltarea bazată pe testare, utilizarea aserțiunilor formale și tratarea excepțiilor.
4. Studentul este capabil să utilizeze clase scrise de alți programatori în dezvoltarea sistemelor.

8. Conținuturi

8.1 Curs	Metode de predare - învățare	Observații ³
1. Elemente de bază ale limbajului C. <ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
2. Programare modulară în C/C++. <ul style="list-style-type: none"> Funcții. Parametri. Fișiere header. Biblioteci. Implementarea modulară a TAD-urilor. Utilizarea pointerului void pentru obținerea genericității. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
3. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++. <ul style="list-style-type: none"> Tipuri de date: vectori și structuri. Tipuri de date: pointeri și referințe. Gestiunea memorie în C/C++ Pointeri la funcții și pointeri spre void. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
4. Programare orientată obiect în C++. <ul style="list-style-type: none"> Clase și obiecte. Membrii unei clase. Modificatori de acces. Constructor/destructori. Gestiunea memoriei în C++ (RAII) Implementarea TAD-urilor în C++ Diagrame UML pentru clase (membri, acces). 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
5. Elemente de programare generica <ul style="list-style-type: none"> Funcții/clase parametrizate. Mecanismul de template din C++ Implementare TAD-uri folosind clase parametrizate Containere și iteratori – biblioteca STL 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
6 Moștenire <ul style="list-style-type: none"> Moștenire simplă. Clase derivate. Principiul substituției. Supraîncărcarea metodelor. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația 	

³ De exemplu aspecte organizatorice, recomandări pentru studenți, aspecte specifice legate de curs/seminar cum ar fi invitarea unor practicieni în domeniu etc.

<ul style="list-style-type: none"> Moștenire multiplă. Relații de specializare/generalizare – reprezentări UML. 	didactică	
7 Polimorfism <ul style="list-style-type: none"> Mostenire, polimorfism Stream-uri I/O. Ierarhia de clase I/O. Formatare. Manipulatori. Fișiere text. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
8 Ierarhii de clase <ul style="list-style-type: none"> Exceptii, spatii de nume Fișiere text. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
9 Interfețe grafice utilizator <ul style="list-style-type: none"> QT Toolkit: instalare, instrumente si module Qt Componente grafice utilizator Layout management 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
10 Elemente de programare bazată pe evenimente <ul style="list-style-type: none"> Evenimente: Semnale si sloturi Qt Proiectare GUI Callback/Observer 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
11 Elemente de programare bazată pe evenimente <ul style="list-style-type: none"> Componente grafice cu modele Șablonul MVC. Studiu de caz. Detalii comenzi – Produse. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
12 Șabloane de proiectare. <ul style="list-style-type: none"> Șabloane de proiectare creaționale, structurale, comportamentale. Exemple 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
13. Aplicație POS (Point Of Sale) <ul style="list-style-type: none"> Șabloane de proiectare Façade, Strategy. Șablonul de proiectare Composite. 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
14. Recapitulare	<ul style="list-style-type: none"> Expunerea interactivă Conversația 	
Bibliografie <ol style="list-style-type: none"> B. Stroustup, The C++ Programming Language, Addison Wesley, 1998. Bruce Eckel, Thinking in C++, www.bruceekel.com Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002 M. Frentiu, B. Parv, Elaborarea programelor. Metode si tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994. E. Horowitz, S. Sahni, D. Mehta, Fundamentals of Data Structures in C++, Computer Science Press, Oxford, 1995. K.A. Lambert, D.W. Nance, T.L. Naps, Introduction to Computer Science with C++, West Publishing Co., New-York, 1996. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca 1996. 		
8.2 Seminar	Metode de predare - învățare	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	<ul style="list-style-type: none"> Expunere interactiva Explicatie 	

(uni și multidimensionali) și structuri.	<ul style="list-style-type: none"> • Conversatie • Exemple • Demonstratie didactica 	
2. TAD de tip container cu elemente generice (void*): reprezentări vizibile și ascunse.	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
3. Clase. Clase simple. Supraîncărcarea operatorilor. Clase cu obiecte ca date membre.	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
4. Clase de tip vector dinamic și iteratori. Moștenire. Polimorfism	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
5. Containeri , iteratori, algoritmi STL	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
6. Interfete grafice utilizator	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
7. Probleme complexe implementate pe baza diagramelor UML. Șabloane de proiectare. Pregătire pentru examenul scris.	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
8.3 Laborator	Metode de predare - învățare	Observații
	<ul style="list-style-type: none"> • 	<ul style="list-style-type: none"> • 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++.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
2. Programare simple în C	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
3. Programare modulara C/C++	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
4. Proces de dezvoltare "feature driven" ,	<ul style="list-style-type: none"> • Lucrare de laborator 	

arhitectura stratificata, test driven development	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea 	
5. Proces de dezvoltare "feature driven", arhitectura stratificata, test driven development	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
6. Programare orientata obiect	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
7. Programare orientata obiect	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
8. Containere, iteratori și algoritmi STL	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
9. Fișiere text	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
10. GUI folosind QT	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
11. GUI folosind QT	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
12 Componente cu modele, MVC, Observer	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
13. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
14. Predare laboratoare (a se vedea observația)	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	



Bibliografie

1. B. Stroustup, The C++ Programming Language, Addison Wesley, 1998.
2. Bruce Eckel, Thinking in C++, www.bruceekel.com
3. Alexandrescu, Programarea moderna in C++. Programare generica si modele de proiectare aplicate, Editura Teora, 2002
4. M. Frentiu, B. Parv, Elaborarea programelor. Metode si 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

9. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
9.4 Curs	<ul style="list-style-type: none"> Corectitudinea și completitudinea cunoștințelor acumulate și capacitatea de a proiecta și implementa programe C++ 	Examen scris (în sesiune)	40%
9.5 Seminar/laborator	<ul style="list-style-type: none"> Capacitatea de a proiecta, testa și depana programe C++ folosind QT 	Evaluare practică (în sesiune)	30%
	<ul style="list-style-type: none"> Corectitudinea programelor C++ și a documentațiilor de laborator 	-documentații -portofoliu -observare continuă	30%
9.6 Standard minim de promovare			
<ul style="list-style-type: none"> 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. 			

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)⁴

Eticheta generală pentru Dezvoltare durabilă								
								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	X
								Nu se aplică nici o etichetă
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Data completării:

06.05.2026

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

⁴ Selectați o singură etichetă, cea care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivește cel mai bine disciplinei. Dacă disciplina tratează tema dezvoltării durabile la modul general (de ex. prin prezentarea/introducerea cadrului general al dezvoltării durabile etc.) atunci se poate alocă eticheta generală de Dezvoltare Durabilă. Dacă niciuna dintre etichete nu descrie disciplina, selectați ultima opțiune: „Nu se aplică nici o etichetă”.

