

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 Matematică și Informatică al Liniei Maghiare
1.4. Domeniul de studii	Matematică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Matematică (în limba maghiară)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare orientată obiect				Codul disciplinei		MLM5006
2.2. Titularul activităților de curs			conf. univ. dr. Darvay Zsolt					
2.3. Titularul activităților de seminar			conf. univ. dr. Darvay Zsolt					
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	obligatoriu - DS	

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	3
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)					35
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					2
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					35
Tutoriat (consiliere profesională)					2
Examinări					6
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				80	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Nu sunt.
4.2. de competențe	Cunoașterea limbajul de programare C.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs dotată cu tablă și videoproiector.
5.2. de desfășurare a seminarului/ laboratorului	Sală de calculatoare, C++.

6. Competențele specifice acumulate

Competențe profesionale/esențiale	- C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic. - C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază. - C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date. - C1.4 Testarea unor aplicații pe baza unor planuri de test. - C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente.
Competențe transversale	- CT1 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. - CT3 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.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	Însușirea conceptelor și principiilor de bază ale programării orientate pe obiecte.
7.2 Obiectivele specifice	- Învățarea elementelor de bază ale limbajului de programare C++. - Recunoașterea diferențelor dintre proiectarea tradițională și cea orientată pe obiecte. - Înțelegerea moștenirii, polimorfismului, legării dinamice și utilizării șabloanelor în vederea dezvoltării unui cod reutilizabil. - Dezvoltarea unor aplicații diferite folosind programare orientată pe obiecte. - Utilizarea structurilor de date și algoritmilor din biblioteca standard. - Utilizarea claselor scrise de alți programatori pentru a rezolva o anumită problemă.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Elemente de bază ale limbajului C ++. - Unități lexicale. Operatori. Conversie. - Tipuri de date. Variabile. Constante. - Vizibilitatea și durata de viață a variabilelor. - Spații de nume.	Prelegerea, dialogul, problematizarea, explicația	
2. Instrucțiuni și funcții în C++. - Instrucțiuni C++. - Declarația și definiția funcției în C++. - Supraîncărcarea funcțiilor. - Funcții inline.	Prelegerea, dialogul, problematizarea, explicația	
3. Tipuri mai complexe decât tipurile de bază și alocarea dinamică a memoriei în C++. - Tablouri și structuri. - Tipul pointer și tipul referință. - Alocarea și eliberarea memoriei în C++. - Pointeri către funcții și funcții void.	Prelegerea, dialogul, problematizarea, explicația	
4. Programare modulară în C++. - Fișiere header. Biblioteci. - Implementarea modulară a tipurilor abstracte de date. - Utilizarea pointerilor void* în vederea generalității.	Prelegerea, dialogul, problematizarea, explicația	

5. Programarea orientată pe obiecte în C++. - Clase și obiecte. - Membrii unei clase. Modificatori de acces. - Constructori și destructori - diagrame de clasă UML (membrii, acces). - Asocieri și agregări într-o diagramă UML.	Prelegerea, dialogul, problematizarea, explicația	
6. Supraîncărcarea operatorilor. Membrii statici. Funcții și clase prietene.	Prelegerea, dialogul, problematizarea, explicația	
7. Moștenire - Clase derivate. Moștenire simplă. - Principiul înlocuirii. - Metode de suprascriere. - Moștenire multiplă. - Relația de specializare / generalizare dintre clase în UML.	Prelegerea, dialogul, problematizarea, explicația	
8. Polimorfismul. - Funcții membru virtuale. - Legare dinamică. - Moștenire virtuală. - Reutilizarea codului (moștenire / compoziție). - Conversii (upcast / downcast).	Prelegerea, dialogul, problematizarea, explicația	
9. Proiectarea orientată pe obiecte și proiectarea bazată pe interfețe. - Clasă abstractă, interfață. - Specificarea interfețelor în UML. - Proiectarea orientată pe obiecte a unei biblioteci de structuri de date.	Prelegerea, dialogul, problematizarea, explicația	
10. Operații de intrare / ieșire. - Stream-uri. Ierarhii de clase de intrare / ieșire. - Formatare. Manipulatori. - Gestionarea fișierelor.	Prelegerea, dialogul, problematizarea, explicația	
11. Programare generică. - Funcții template. - Clase template. - Reutilizarea codului.	Prelegerea, dialogul, problematizarea, explicația	
12. Biblioteca standard C++ (STL) - Containere și iteratori. - Algoritmi standard.	Prelegerea, dialogul, problematizarea, explicația	
13. Excepții. - Conceptul de excepție. - Generarea și tratarea excepțiilor în C++.	Prelegerea, dialogul, problematizarea, explicația	
14. Mediul de dezvoltare Visual C++. - Elemente ale interfeței grafice cu utilizatorul.	Prelegerea, dialogul, problematizarea, explicația	
Bibliografie 1. A. Alexandrescu, Programarea modernă în C++. Programare generică și modele de proiectare aplicate, Editura Teora, 2002. 2. R. Andonie, I. Garbacea, Algoritmi fundamentali. O perspectiva C++, Editura Libris, Cluj_Napoca, 1995. 3. V. Cioban, Zs. Darvay, Metode evaluate de programare, UBB-Mate_Info, 1999. 4. B. Eckel, Thinking in C++, (second edition), vol. 1 & 2, 2000, 2004. 5. M. Frentiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994. 6. B. Milewski, C++ in Action, 1997. 7. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca, 1996.		
8.2 Seminar / laborator	Metode de predare	Observații
S1. Pointeri către funcții. Tipuri abstracte de date. Programare modulară	exerciții, muncă individuală, dialog	
S2. Spațiu de nume. Clasă cu mai mulți constructori.	exerciții, muncă individuală, dialog	
S3. Dată membru de tip clasă. Excepții.	exerciții, muncă individuală, dialog	
S4. Clase template. Supraîncărcarea operatorilor.	exerciții, muncă individuală, dialog	

S5. Funcții membru virtuale. Clase abstracte.	exerciții, muncă individuală, dialog, proiect practic	
S6. Conversii.	exerciții, muncă individuală, dialog, proiect practic	
S7. Implementarea unui tip abstract de date și a unui iterator corespunzător.	exerciții, muncă individuală, dialog	
L1. Exerciții introductive în limbajul de programare C++.	exerciții, muncă individuală, dialog	
L2. Probleme C++ simple folosind funcții.	exerciții, muncă individuală, dialog	
L3. Protecția datelor cu ajutorul programării modulare în C. Spații de nume în C++.	exerciții, muncă individuală, dialog	
L4. Probleme de bază privind noțiunea de clasă. Dată membru de tip clasă.	exerciții, muncă individuală, dialog	
L5. Exerciții folosind excepții.	exerciții, muncă individuală, dialog	
L6. Exerciții cu funcții și clase template.	exerciții, muncă individuală, dialog	
L7. Exerciții privind supraîncărcarea operatorilor.	exerciții, muncă individuală, dialog	
L8. Implementarea înlănțuită a structurilor de date. Conversii implicite.	exerciții, muncă individuală, dialog	
L9. Conversii explicite. Funcții membru virtuale.	exerciții, muncă individuală, dialog	
L10. Exerciții cu clase virtuale și abstracte.	exerciții, muncă individuală, dialog	
L11. Implementarea unui tip abstract de date și a unui iterator corespunzător.	exerciții, muncă individuală, dialog	
L12. Utilizarea containerelor din STL.	exerciții, muncă individuală, dialog	
L13. Utilizarea algoritmilor din STL.	exerciții, muncă individuală, dialog	
L14. Evaluarea proiectelor.	exerciții, muncă individuală, dialog, proiect practic	
Bibliografie 1. B. Parv, A. Vancea, Fundamentele limbajelor de programare, Microinformatica, Cluj-Napoca, 1996. 2. B. Stroustrup, The C++ Programming Language, Addison Wesley, 1997. 3. B. Stroustrup: The C++ Programming Language Special Edition, AT&T, 2000.		

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

- Conținutul cursului este în concordanță cu cursuri asemănătoare predate în țară sau în străinătate, care se referă la programare orientată pe obiecte și limbajul de programare C++.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Activitate la curs și cunoștințe teoretice	Test teoretic	40%
10.5 Seminar/laborator	Activitatea la seminar	Exerciții la seminar	10%
	Activitatea la laborator	Exerciții la laborator și teme pentru acasă	15%
	Crearea de aplicații complexe	Proiecte	10%
	Însușirea cunoștințelor practice	Examen practic	25%

10.6 Standard minim de performanță

- Cunoașterea elementelor de bază ale limbajelor de programare C și C ++. Recunoașterea diferențelor dintre cele două limbaje.
- Cunoașterea conceptelor legate de programarea orientată pe obiecte.
- Exploatarea capacității orientate pe obiecte a limbajului C++ în cazul dezvoltării unor aplicații.
- La examenul teoretic și cel practic trebuie obținut cel puțin 50%.

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

Nu se aplică.

Data completării:

20.09.2025

Semnătura titularului de curs

Conf. univ. dr. Darvay Zsolt

.....

Semnătura titularului de seminar

Conf. univ. dr. Darvay Zsolt

.....

Data avizării în departament:

20.09.2025

Semnătura directorului de departament

Conf. univ. dr. András Szilárd

.....