FIŞA DISCIPLINEI

Programare orientată obiect

Anul universitar 2025-2026

1. Date despre program

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

2. Date despre disciplină

2.1. Denumirea disciplinei	**Bazele programării orientate obiect**					Codul disciplinei	**MLE5234**
2.2. Titularul activităților de curs	**Lect. dr. ing. Diana Laura Borza**						
2.3. Titularul activităților de seminar	**Lect. dr. ing. Diana Laura Borza**						
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligatorie

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)					24
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					19
Tutoriat (consiliere profesională)					9
Examinări					13
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	Fundamentele Programării Bazele Matematice ale Calculatoarelor
4.2. de competențe	Deprinderi medii de programare în limbaje de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Studenții vor participa la curs cu telefoanele mobile închise Studenții vor participa la curs cu laptop-urile închise; studenții cu nevoi speciale vor discuta aceste chestiuni la începutul semestrului
5.2. de desfășurare a seminarului/ laboratorului	Studenții vor participa la laborator cu telefoanele mobile închise Laborator cu calculatoare; medii de programare cu un compilator C++, librăria Qt instalată, doxygen, git

6.. Competențele specifice acumulate[1]

Competențe profesionale/esențiale	• prelucrarea matematică a datelor, analiza şi interpretarea unor fenomene şi procese • elaborarea şi analiza unor algoritmi pentru rezolvarea problemelor • conceperea modelelor matematice pentru descrierea unor fenomene • programarea în limbaje de nivel înalt • analiza, testarea şi utilizarea sistemelor informatice
Competențe transversale	• aplicarea regulilor de muncă riguroasă şi eficientă, manifestarea unor atitudini responsabile faţă de domeniul didactico-ştiinţific, pentru valorificarea optimă şi creativă a propriului potenţial în situaţii specifice, cu respectarea principiilor şi a normelor de etică profesională • desfăşurarea eficientă şi eficace a activităţilor organizate în echipă • utilizarea eficientă a surselor informaţionale şi a resurselor de comunicare şi formare profesională asistată, atât în limba română, cât şi într-o limbă de circulaţie internaţională

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Să înțeleagă conceptele paradigmei de programare orientată obiect și să proiecteze soluții ale problemelor la scară mică/medie, folosind C++ și Qt.
7.2 Obiectivele specifice	• Să înțeleagă diferențele dintre programarea imperativă tradițională și programarea orientate obiecte. • Să înțeleagă noțiunea de clasă ca elemente fundamentale, modulare. • Să înțeleagă rolul moștenirii, polimorfismului, legării dinamice și structurilor generice în construirea codului reutilizabil. • Să proiecte interfețe utilizator și să scrie programe C++ la scară mică/medie folosind Qt. • Să folosească clase scrise de alți programatori și biblioteci terțe atunci când își construiesc sistemele.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. **Introducere în C++** (elementele de bază ale limbajului de programare C/C++, tipuri de date, variabile, constante, durata de viață a variabilelor, instrucțiuni, funcții: declarație și definiție, funcții de supraîncărcare).	• Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică	
2. **Programare modulară în C/C++** (funcții, parametri formali și actuali, pointeri și managementul memoriei, stiva și *heap*, pointeri către funcții, fișiere *header*, programare modulară, biblioteci).	• Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică	
3. **Programare orientată obiect în C++** (introducere în programarea orientată pe obiecte, caracteristicile programării orientată obiect, abstractizare, încapsulare, clase și obiecte, modificatori de acces, crearea și distrugerea obiectelor, supraîncărcarea operatorilor, elemente statice și *friend*).	• Expunerea interactivă • Explicaţia • Conversaţia • Demonstraţia didactică	
4. **Moștenire și polimorfism** (clase de bază și derivate, principiul substituției	• Expunerea interactivă • Explicaţia	

[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.

Liskov, suprascrierea metodei, moștenirea și polimorfismul).	• Conversația • Demonstrația didactică	
5. **Polimorfism** (legare statică și dinamică, metode virtuale, moștenire multiplă, upcasting și downcasting, clase abstracte, diagrame de clasă UML și relații între clase).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
6. **Programare generică în C++. Librăria standard C++ (C++ Standard Template Library)** (șabloane de funcție (*function templates*), șabloane de clasă (*class templates*), containere în STL: tablou static, vector, listă, stivă, heap, *map*, mulțime), iteratoare, algoritmi STL, funcții lambda).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
7. **Streams-uri și tratarea excepțiilor** (fluxuri de ieșire de intrare, operatori de inserare și extracție, operatori de inserare și extracție de supraîncărcare, formatare, manipulatoare, fișiere text, gestionarea excepțiilor,*exception safe code*).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
8. **Managementul resurselor și RAII** (Resource Acquisition Is Initialization (RAII), *smart pointers*, *move semantics,* std::unique_ptr, std::shared_ptr, std::weak_ptr)	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
9. **Interfețe grafice în Qt** (Qt Toolkit –descriere generală, component grafice Qt, Qt Designer).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
10. **Programare bazată pe evenimente I** (callback-uri, evenimente, semnale și sloturi în Qt).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
11. **Programare bazată pe evenimente II** (Model View Controller, Models și Views în Qt).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
12. **Șabloane de proiectare I** (șabloane de proiectare creaționale, structurale, comportamentale ; exemple; singleton, factory method, adapter).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
13. **Șabloane de proiectare II** (façade, observer, strategy, studii de caz și exemple).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
14. **Recapitulare** (revizuire a celor mai importante subiecte abordate de curs, ghid de examinare).	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	

Bibliografie

1. B. Stroustrup. ***The C++ Programming Language***, Addison Wesley, 1998.
2. Bruce Eckel. ***Thinking in C++***, Prentice Hall, 1995.
3. A. Alexandrescu. ***Programarea modernă în C++: Programare generică și modele de proiectare aplicate***, Editura Teora, 2002.
4. S. Meyers. ***Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition)***, Addison-Wesley, 2005.
5. S. Meyers. More effective C++: 35 New Ways to Improve Your Programs and Designs, Addison-Wesley, 1995.
6. B. Stroustrup. ***A Tour of C++***, Addison-Wesley, 2013.
7. C++ reference (http://en.cppreference.com/w/).

8. Qt Documentation (http://doc.qt.io/qt-5/).
9. E. Gamma, R. Helm, R. Johnson, J. Vlissides. ***Design Patterns: Elements of Reusable Object-Oriented Software***, Addison-Wesley Longman Publishing, 1995.

8.2 Seminar / laborator	Metode de predare	Observații
Seminar		
1. Probleme simple în C, funcții, structure, tablouri, tipul de date enumerare (*enum*)	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
2. Programare modulară	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
3. Clase și obiecte.Supraîncărcarea operatorilor.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
4. Moștenire. Polimorfism. *Templates*.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
5. Fișiere, excepții. Containeri STL, iteratori, algoritmi.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
6. Interfețe grafice utilizator.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
7. Implementarea pe baza diagramelor de clasă. Șabloane de proiectare.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	

Bibliografie

1. B. Stroustrup. ***The C++ Programming Language***, Addison Wesley, 1998.
2. Bruce Eckel. ***Thinking in C++***, Prentice Hall, 1995.
3. A. Alexandrescu. ***Programarea modernă în C++: Programare generică și modele de proiectare aplicate***, Editura Teora, 2002.
4. S. Meyers. ***Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition)***, Addison-Wesley, 2005.
5. S. Meyers. More effective C++: 35 New Ways to Improve Your Programs and Designs, Addison-Wesley, 1995.
6. B. Stroustrup. ***A Tour of C++***, Addison-Wesley, 2013.
7. C++ reference (http://en.cppreference.com/w/).
8. Qt Documentation (http://doc.qt.io/qt-5/).
9. E. Gamma, R. Helm, R. Johnson, J. Vlissides. ***Design Patterns: Elements of Reusable Object-Oriented Software***, Addison-Wesley Longman Publishing, 1995.

Laborator		
1. Instalarea unui compilator C++, IDE. Noțiuni de bază C/C++.	• Explicația • Conversația	
2. Probleme introductive (în C).	• Explicația • Conversația	
3. Programare modulară	• Explicația • Conversația	
4. Clase și obiecte în C++. Constructorul de copiere, operatorul de asignare, destructori.	• Explicația • Conversația	
5. Moștenire. Suprascrierea metodelor.	• Explicația • Conversația	
6. Moștenire și polimorfism, funcții virtuale.	• Explicația • Conversația	

7. Test de laborator.	Test de laborator	
8. Containeri STL, iteratori, algoritmi.	• Explicația • Conversația	
9. Tratarea excepțiilor.	• Explicația • Conversația	
10. Fișiere. Testarea programelor.	• Explicația • Conversația	
11. Interfețe grafice în Qt I.	• Explicația • Conversația	
12. Interfețe grafice în Qt II. Semnale și slot-uri Qt.	• Explicația • Conversația	
13. Șabloane de proiectare.	• Explicația • Conversația	
14. Test de laborator.	Test de laborator	

Bibliografie

1. B. Stroustrup. ***The C++ Programming Language***, Addison Wesley, 1998.
2. R. Gilberg. **C++ Programming: An Object-Oriented Approach**, McGraw-Hill Education, 2019
3. A. Alexandrescu. ***Programarea modernă în C++: Programare generică și modele de proiectare aplicate***, Editura Teora, 2002.
4. S. Meyers. ***Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition)***, Addison-Wesley, 2005.
6. B. Stroustrup. ***A Tour of C++,*** Addison-Wesley, 2013.
7. C++ reference (http://en.cppreference.com/w/).
8. Qt Documentation (http://doc.qt.io/qt-5/).
9. E. Gamma, R. Helm, R. Johnson, J. Vlissides. ***Design Patterns: Elements of Reusable Object-Oriented Software***, Addison-Wesley Longman Publishing, 1995.
10. Bruce Eckel. ***Thinking in C++***, Prentice Hall, 1995.

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

- The course respects the ACM Curricula Recommendations for Computer Science studies.
- The course exists in the studying program of all major universities in Romania and abroad.
- The content of the course is considered by the software companies as important for average object-oriented programming skills.

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, capacitatea de a proiecta și implementa programe corecte C++.	Evaluare scrisă în sesiune	**60%**
10.5 Seminar/laborator	Abilitatea de a proiecta, implementa, testa și depana un program C++ cu o interfață grafică cu utilizatorul.	Evaluare practică, două teste în timpul semestrului	**20%**
	Proiect	Proiectarea, implementarea și testarea unei aplicații mici-medii care utilizează o arhitectură pe 3 niveluri. Documentație.	**20%**
10.6 Standard minim de performanță			
Cunoașterea elementelor fundamentale de teorie, rezolvarea unei aplicații simple într-un limbaj orientat obiect. Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoaștere și înțelegere a domeniului, că			

este capabil să exprime cunoştinţele într-o formă coerentă, că are capacitatea de a stabili anumite conexiuni şi de a utiliza cunoştinţele în rezolvarea unor probleme.

- Pentru promovare sunt necesare următoarele criterii minimale: nota minim 5 (fără rotunjire) la lucrarea scrisă şi nota finală minim 5 la proiect și testele de laborator(fără rotunjire).
- Examenul scris este organizat în două părți: o probă teoretică și o probă practică. Studenții trebuie să obțină o notă mai mare de 5 la ambele probe.
- Pentru promovare, prezența este obligatorie pentru minim 5 seminarii şi 12 laboratoare.

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

Nu se aplică.

Data completării:	Semnătura titularului de curs	Semnătura titularului de seminar
27 aprilie 2025	**Lect. dr. ing. Diana Laura Borza**	**Lect. dr. ing. Diana Laura Borza**

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.*".