

FIȘA DISCIPLINEI

Programare logică și funcțională

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babes-Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclu de studii	Licență
1.6. Programul de studii / Calificarea	Informatică în limba germană
1.7. Forma de învățământ	IF

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare logică				Codul disciplinei	MLG5143
2.2. Titularul activităților de curs			Conf. Dr. Christian Săcărea				
2.3. Titularul activităților de seminar			Lect. Dr. Albișoru Florin				
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	C	2.7. Regimul disciplinei	Obligatorie (DF)

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

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/practic	1+1
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					30
Tutoriat (consiliere profesională)					10
Examinări					14
Alte activități					-
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				94	
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, Logică computațională, Structuri de date și algoritmi
4.2. de competențe	-

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 limbajele de programare GCLisp, CLisp și TurboProlog

6.1. Competențele specifice acumulate¹

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.

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 • C1.5 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

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: • metodele, algoritmi, paradigmele 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, paradigmele 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. • 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 paradigma programării declarative (programarea funcțională și programarea logică).
7.2 Obiectivele specifice	• Să introducă câte un limbaj de programare pentru fiecare din aceste paradigme (Common Lisp și Prolog). • Să inducă ideea utilizării acestor paradigme în funcție de necesitățile aplicațiilor. • Să asigure baza necesară urmăririi unor cursuri avansate de programare declarativă.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
----------	-------------------	------------

1. Programare și limbaje de programare. Programare imperativa vs. programare declarativă. Introducere. Recursivitate. Exemple.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
2. Elemente fundamentale ale limbajului Prolog. Fapte și reguli Prolog. Întrebări. Strategia de control în Prolog. Variabile și propoziții compuse. Variabile anonime. Reguli de definire a potrivirilor. Model de flux. Secțiunile unui program Prolog. Exemple.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
3. Programul Prolog. Domenii predefinite. Întrebări interne și externe. Predicate cu aritate multiplă. Simbolul IF (Prolog) și instrucțiunea IF (alte limbaje). Directive de compilare. Expresii aritmetice și comparații. Operații de intrare / ieșire. Șiruri de caractere.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
4. Backtracking. Controlarea backtracking-ului. Predicatele fail și ! (cut). Utilizarea lui !. Tipuri de tăieturi. Predicatul "not". Liste Prolog. Recursivitate. Exemple de tratare a backtracking-ului. Găsirea tuturor soluțiilor în același timp. Exemple de predicate Prolog. Predicate nedeterminate	Expunerea interactivă Explicația Conversația Demonstrația didactică	
5. Obiecte compuse și functori. Unificarea obiectelor compuse. Argumente de tipuri multiple; liste eterogene. Compararea obiectelor compuse. Backtracking cu ciclări. Exemple de proceduri recursive. Cadrul stivei. Optimizarea prin recursivitate de coadă. Utilizarea tăieturii pentru păstrarea recursivității de coadă.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
6. Structuri de date recursive. Arborii ca structuri de date. Construirea și traversarea unui arbore. Arbori de căutare. Baza de date internă a sistemului Prolog. Secțiunea database. Declararea bazei de date interne. Predicate relativ la operații cu baza de date internă.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
7. Examen scris PROLOG	Lucrarea scrisă	
<i>Programare Funcțională. Limbajul LISP</i>		
8. Importanța programării funcționale ca noua metodologie de programare. Istoric și prezentare a limbajului LISP. Elemente de bază Lisp. Structuri dinamice de date. Reguli sintactice și semantice. Clasificarea funcțiilor Lisp. Funcții primitive în Lisp.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
9. Predicate de bază în Lisp. Predicate pentru liste; pentru numere. Funcții logice și aritmetice. Definirea funcțiilor utilizator. Ramificarea prelucrărilor. Metoda variabilei colectoare. Exemple	Expunerea interactivă Explicația Conversația Demonstrația didactică	
10. Gestiunea simbolurilor. Alte funcții de acces la liste. OBLIST și ALIST. Funcții cu caracter destructiv. Comparații. Alte funcții interesante. Exemple.	Expunerea interactivă Explicația Conversația Demonstrația didactică	

11. Mecanisme definiționale evaluate Forma EVAL. Forme funcționale; funcțiile FUNCALL si APPLY. Expresii LAMBDA. Expresii LABEL. Exemple	Expunerea interactivă Explicația Conversația Demonstrația didactică	
12. Generatori, argumente funcționale. Funcții MAP. Forme iterative. Exemple.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
13. Alte elemente ale limbajului Lisp. Structuri de date. Macrodefiniții. Argumente opționale. Exemple.	Expunerea interactivă Explicația Conversația Demonstrația didactică	
14. Examen scris LISP	Lucrare scrisă	

Bibliografie în limba germană

- Goos, G., Zimmermann, W., Vorlesungen über Informatik, Band 1, Grundlagen und funktionales Programmieren, Springer, 2006.
- Lippe, W-M., Funktionale und Applikative Programmierung, Springer 2009.
- Cordes/Kruse/Langendorfer/Rust: Prolog — Eine methodische Einführung Vieweg, 1990, 2. Auflage, 245 Seiten
- Hanus: Problemlos mit Prolog Teubner, 1987, 2. Auflage, 224 Seiten
- Kleine Buning/Schmitgen: Prolog, Teubner, 1988, 2. Auflage, 311 Seiten

In alte limbi

- Czibula, G., Pop, H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastra, Cluj-Napoca, 2012
- Pop, H.F., Șerban, G., Programare în Inteligența Artificială - Lisp și Prolog, Editura Albastră, Cluj-Napoca, 2003
- <http://www.lpa.co.uk>, Logic Programming
- Field A., Functional Programming, Addison Wesley, New York, 1988.
- Winston P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.

8.2 Seminar / laborator	Metode de predare	Observații
		Laboratorul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
Lab 1/Sem1: Recursivitate	Lucrare de laborator Explicația Conversația Modelarea	Se predă tema primită la Lab 1 Se primește tema pentru Lab 2
Lab 2/Sem 2: Liste în Prolog	Lucrare de laborator Explicația Conversația Modelarea	Se predă tema primită la Lab 2 Se primește tema pentru Lab 3
Lab 3/Sem3: Arbori în Prolog. Gestiunea listelor în Prolog.	Lucrare de laborator Explicația Conversația Modelarea	Se predă tema primită la Lab3 Se primește tema pentru Lab 4
Lab 4/Sem4: Backtracking în Prolog	Lucrare de laborator Explicația Conversația Modelarea	
Lab 4/Sem4: Proba practică Prolog	Lucrare practică	1 oră

Lab 5/Sem 5: Programare recursivă în Lisp	Lucrare de laborator Explicația Conversația Modelarea	Se predă tema primită la Lab 5 Se primește tema pentru Lab 6
Lab 6/Sem 6: Folosirea funcțiilor MAP	Lucrare de laborator Explicația Conversația Modelarea	Se predă tema primită la Lab 6 Se primește tema pentru Lab 7
Lab 7/Sem 7: Programare iterativă în Lisp	Lucrare de laborator Explicația Conversația Modelarea	1 oră: Se predă tema primită pentru Lab 7
Lab 7/Sem 7: Proba practică Lisp	Lucrare practică	1 oră

Bibliografie în limba germana

- Hölldobler, S., Logik und Logikprogrammierung: Synchron Wissenschaftsverlag der Autoren, Heidelberg, 2001
- Weisweber, W., Logische Programmierung in der Praxis, Redline GmbH, 2000
- Lippe, W-M., Funktionale und Applikative Programmierung, Springer 2009.

Bibliografie generala:

- Czibula, G., Pop, H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastră, Cluj-Napoca, 2012
- Documentația produselor: Gold Common Lisp 1.01 și 4.30, XLisp, Free Lisp.
- <http://www.swi-prolog.org/>

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

- Conținutul disciplinei este în concordanță cu ceea ce se face în alte centre universitare din țară și din străinătate.
- Conținutul disciplinei asigură cunoștințele fundamentale necesare pentru programare în Lisp și Prolog la eventualii angajatori.

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 asimilate. • Gradul de asimilare a limbajului de specialitate	Evaluare scrisă (în timpul semestrului): examen parțial Prolog (curs 7)	30%
		Evaluare scrisă (în timpul semestrului): examen parțial Lisp (curs 14)	30%
10.5 Laborator	• Implementarea în Lisp și Prolog a problemelor de laborator • Redactarea documentației de laborator • Respectarea termenelor de predare.	Documentații și programe	10%
		Proba practică Prolog (1 oră, lab. 4)	15%
		Proba practică Lisp (1 oră, lab. 7)	15%

10.6 Standard minim de performanță

- Cunoașterea elementelor fundamentale de teorie, rezolvarea unei aplicații simple într-un limbaj declarativ. 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.

- Pentru promovare sunt necesare următoarele criterii minime: nota minim 4 la fiecare din lucrările scrise, predarea a minim 5 lucrări de laborator și nota finală minim 5.

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

	Eticheta generală pentru Dezvoltare durabilă							

Data completării:
11.04.2025

Semnătura titularului de curs
Conf. Dr. Christian Săcărea

Semnătura titularului de seminar
Lect. Dr. Albișoru Florin

Data avizării în departament:
25.04.2025

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*".