

FIȘA DISCIPLINEI

PROGRAMARE FUNCȚIONALĂ ȘI LOGICĂ

Anul universitar 2026/2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Matematică și Informatică
1.3. Departamentul	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 logică și funcțională			Codul disciplinei	MLE5009
2.2. Titularul activităților de curs	Prof. dr. Horia F. Pop				
2.3. Titularul activităților de seminar	Prof. dr. Horia F. Pop				
2.4. Anul de studiu	2	2.5. Semestrul	3	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	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	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)					22
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					18
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					27
Tutoriat (consiliere profesională)					11
Examinări					26
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	MLE5005: Programming Fundamentals MLE5055: Computational Logic MLE5022: Data Structures and Algorithms
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	Studentii vor participa la curs cu telefoanele mobile închise Studentii 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	Studentii vor participa la laborator cu telefoanele mobile închise Laborator cu calculatoare; medii de programare în limbaje declarative de nivel înalt (CLisp, SWIProlog)

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
CP2	Remediază erorile din software
Competențe transversale	
Codul competenței	Competență
CT1	Lucrează independent
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	Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.	Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
CP4	Studentul/absolventul alege, explică și specifică fundamentele matematice aplicate în informatică, inclusiv logica formală, algebra, probabilitățile și statisticile.	Studentul/absolventul aplică, evaluează, propune metodele matematice pentru modelarea, simularea și rezolvarea problemelor informatice.
CP5	Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și framework-uri actuale.	

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Să deprindă studentul cu paradigma programării declarative (programarea funcțională și programarea logică).
2. Să introducă câte un limbaj de programare pentru fiecare din aceste paradigme (Common Lisp și SWI Prolog).
Abilități academice specifice (Specific academic skills)
1. Să inducă ideea utilizării acestor paradigme în funcție de necesitățile aplicațiilor.
2. Să asigure baza necesară urmării unor cursuri avansate.
3. Abilitatea de a aplica tehnici de programare declarativă la diferite probleme din viața reală.
4. Capacitatea de a modela fenomene folosind tehnici declarative

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.

5. Abilități îmbunătățite de programare folosind paradigma declarativă

8. Conținuturi



















8.1 Curs	Metode de predare	Observații
Elemente de bază din Prolog. Fapte și reguli în Prolog. Goluri. Strategia de control din Prolog. Variabile și propoziții compuse. Variabile anonime. Reguli pentru potrivire. Modelul de flux. Secțiuni ale unui program Prolog. Exemple	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Programul Prolog. Domenii predefinite. Obiective interne și externe. Predicate de aritate multiple. Simbolul IF (Prolog) și instrucțiunea IF (alte limbaje). Directivele compilatorului. Expresii aritmetice și comparații. Operațiuni de intrare/ieșire	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Înapoi. Controlul backtracking-ului. "Eșecul" și "!" (tăietura). Folosind predicatul "!". Tipuri de tăieturi. Predicatul "not". Liste în Prolog. Recursivitate. Exemple de backtracking în Prolog. Găsirea tuturor soluțiilor în același timp. Exemple de predicate în Prolog. Predicate nedeterministe	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Obiecte compuse și functori. Unificarea obiectelor compuse. Argumente de mai multe tipuri; liste eterogene. Comparații pentru obiecte compuse. Înapoi cu cicluri. Exemple de proceduri recursive. Cadrul de tip stack. Optimizare folosind "recursivitatea cozii". Folosirea predicatului "cut" pentru a păstra "recursivitatea cozii".	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Structuri de date recursive. Arbori ca structuri de date. Crearea și traversarea unui copac. Arbori de căutare. Baza de date internă a Prolog. Secțiunea "bază de date". Declarația bazei de date interne. Predicate referitoare la operațiuni cu baza de date internă.	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Probleme avansate ale Backtracking în Prolog. Gestionarea fișierelor în Prolog.	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Programare și limbaje de programare. Programare imperativă vs. programare declarativă. Introducere. Importanța programării funcționale ca o nouă metodologie de programare. Istoria și prezentarea Lisp	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Elemente de bază în Lisp. Structuri dinamice de date. Reguli sintactice și semantice. Clasificarea funcțiilor în Lisp. Funcții primitive în Lisp. Predicate de bază în Lisp.	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Predicate pentru liste; pentru numere. Logică și funcții aritmetice. Definirea funcțiilor utilizatorului. Forma condiționată. Metoda variabilelor de colectare. Exemple	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Managementul simbolurilor. Alte funcții pentru accesarea listelor. OBLIST și ALIST. Funcții distructive. Comparații. Alte funcții interesante. Exemple	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Mecanisme de definiție. Forma EVAL. Forme funcționale; funcțiile FUNCALL și APPLY. Expresii LAMBDA, expresii LABEL. Generatoare, argumente funcționale. Funcții	Expunere: descriere, explicație, exemple, discuții și studii de caz	

MAP. Forme iterative. Exemple		
Alte elemente din Lisp. Structuri de date. Macro-definiții. Argumente opționale. Exemple	Expunere: descriere, explicație, exemple, discuții și studii de caz	
Bibliografie 1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, ClujNapoca, 2003 3. http://www.ifcomputer.com/PrologCourse , Lecture on Prolog 4. http://www.lpa.co.uk , Logic Programming 5. FIELD A., Functional Programming, Addison Wesley, New York, 1988. 6. WINSTON P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.		
8.2 Seminar	Metode de predare	Observații
S1. Recursivitate	Explicația; Conversația; Modelarea; Studiu de caz	
S2. Liste în Prolog	Explicația; Conversația; Modelarea; Studiu de caz	
S3. Prelucrarea listelor eterogene	Explicația; Conversația; Modelarea; Studiu de caz	
S4. Backtracking în Prolog	Explicația; Conversația; Modelarea; Studiu de caz	
S5. Prelucrarea listelor în LISP	Explicația; Conversația; Modelarea; Studiu de caz	
S6. Funcții MAP în LISP	Explicația; Conversația; Modelarea; Studiu de caz	
S7. Recapitulare	Explicația; Conversația; Modelarea; Studiu de caz	
Bibliografie 1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, ClujNapoca, 2003 3. Product documentation: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp, CLisp. 4. Product documentation: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog, SWI Prolog.		
8.3 Laborator	Metode de predare	Observații
Lab 1: Recursivitate în Pseudocod	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 1 și verificată la lab 1
Lab 2: Liste în Prolog	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 1 și verificată la lab 2
Lab 3: Arbori în Prolog. Gestiunea listelor (eterogene) în Prolog.	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 2 și verificată la lab 3
Lab 4: Backtracking în Prolog	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 3 și verificată la lab 4
Lab 4: Proba practică Prolog	Lucrare practică	1 oră
Lab 5: Programare recursivă în Lisp	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 4 și verificată la lab 5
Lab 6: Programare recursivă în Lisp	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 5 și verificată la lab 6
Lab 7: Folosirea funcțiilor MAP.	Lucrare de laborator; Explicația; Conversația; Modelarea	Problemă dată la lab 6 și verificată la lab 7
Lab 7: Proba practică Lisp	Lucrare practică	1 oră
Bibliografie 1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, ClujNapoca, 2003 3. Product documentation: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp, CLisp. 4. Product documentation: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog, SWI Prolog.		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Corectitudinea și completitudinea cunoștințelor asimilate. Gradul de asimilare a limbajului de specialitate.	Evaluare scrisă: test Prolog+Lisp (curs 13+14)	60%
9.5 Seminar/laborator	Activitatea din timpul seminariilor	Evaluare a activității la seminarii	BONUS 5%
	Implementarea în Lisp și Prolog a conceptelor și algoritmilor prezentați la curs 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%
9.6 Standard minim de performanță			
<p>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.</p> <p>Pentru promovare este OBLIGATORIE prezența la minim 5 seminarii și 6 laboratoare. Pentru promovare sunt necesare următoarele criterii minimale: nota minim 5 la lucrarea scrisă și nota finală minim 5. Activitatea din timpul semestrului nu se poate reface în sesiuni. Toate cerințele pentru examenul de restanță sunt identice cu cele de mai sus.</p>			

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)³

 <input type="radio"/> Eticheta generală pentru Dezvoltare durabilă								
								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
								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"/>

³ 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”.

Data completării:
10/5/2026

Semnătura titularului de curs

Prof. dr. Horia F. Pop

Semnătura titularului de seminar

Prof. dr. Horia F. Pop

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA