

FIȘA DISCIPLINEI

Programare logică și funcțională

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 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	MLR5009
2.2. Titularul activităților de curs		Prof. dr. CZIBULA Gabriela					
2.3. Titularul activităților de seminar		Prof. dr. CZIBULA Gabriela					
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	C	2.7. Regimul disciplinei	Obligativu

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	1 sem + 1 lab
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					14
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					27
Tutoriat (consiliere profesională)					15
Examinări					16
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
4.2. de competențe	Abilități medii de programare

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 CLisp și SWIProlog

6.1. Competențele specifice acumulate¹

Competențe profesionale/esențiale	• utilizarea bazelor teoretice ale informaticii și a modelelor formale • programarea în limbaje de nivel înalt
Competențe transversale	• 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 • 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, paradigme 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, paradigme 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 SWI Prolog). • Să inducă ideea utilizării acestor paradigme în funcție de necesitățile aplicațiilor. • Să asigure baza necesară urmăririi unor cursuri avansate de programare declarativă.

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.

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.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
<i>Programare Funcțională. Limbajul LISP</i>		
7. 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	
8. 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	
9. 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	
10. Mecanisme definiționale evoluate Forma EVAL. Forme funcționale; funcțiile FUNCALL și APPLY. Expresii LAMBDA. Expresii LABEL.	• Expunerea interactivă • Explicația • Conversația	

Exemple	• Demonstrația didactică	
11. Generatori, argumente funcționale. Funcții MAP. Forme iterative. Exemple.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
12. Alte elemente ale limbajului Lisp. Structuri de date. Macrodefiniții. Argumente opționale. Exemple.	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	
13,14. Colocviu (examen scris) PROLOG+LISP	• Expunerea interactivă • Explicația • Conversația • Demonstrația didactică	

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, Cluj-Napoca, 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
		Seminarul este structurat sub forma a 2 ore din 2 în 2 săptămâni.
S1. Recursivitate	• Explicația • Conversația • Modelarea • Exercițiul	
S2. Liste în Prolog	• Explicația • Conversația • Modelarea • Exercițiul	
S3. Prelucrarea listelor eterogene	• Explicația • Conversația • Modelarea • Exercițiul	
S4. Backtracking în Prolog	• Explicația • Conversația • Modelarea • Exercițiul	
S5. Prelucrarea listelor în LISP	• Explicația • Conversația • Modelarea • Exercițiul	
S6. Funcții MAP în LISP	• Explicația • Conversația • Modelarea • Exercițiul	
S7. Recapitulare	• Explicația • Conversația • Modelarea • Exercițiul	

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. Documentatia produsului GNU Common Lisp: <https://www.gnu.org/software/gcl/>.
3. Documentatia produsului SWI-Prolog: <http://www.swi-prolog.org/>

8.3 Laborator	Metode de predare	Observații
		• Laboratorul este structurat sub forma a 2 ore din 2 în 2 săptămâni. • Exceptând Lab 1 care se predă la finalul laboratorului, restul temelor de laborator se predau în laboratorul următor primirii temei,
Lab 1: Recursivitate	• Lucrare de laborator • Explicația • Conversația • Modelarea	
Lab 2: Liste în Prolog	• Lucrare de laborator • Explicația • Conversația • Modelarea	
Lab 3: Gestiunea listelor (eterogene) în Prolog.	• Lucrare de laborator • Explicația • Conversația • Modelarea	
Lab 4: Backtracking în Prolog	• Lucrare de laborator • Explicația • Conversația • Modelarea	
Lab 4: Proba practică Prolog	Lucrare practică	1 oră
Lab 5: Programare recursivă în Lisp	• Lucrare de laborator • Explicația • Conversația • Modelarea	
Lab 6: Folosirea funcțiilor MAP.	• Lucrare de laborator • Explicația • Conversația • Modelarea	
Lab 4: Proba practică Prolog	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. Documentatia produsului GNU Common Lisp: <https://www.gnu.org/software/gcl/>.
3. Documentatia produsului SWI-Prolog: <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	Evaluare scrisă: examen scris Prolog+Lisp (curs 14)	60%

	completitudinea cunoștințelor asimilate. • Gradul de asimilare a limbajului de specialitate.		
10.5 Seminar	• Activitatea din timpul seminariilor	Evaluare a activității la seminarii – se acordă un bonus de max. 0.5p pentru activitatea din timpul seminariilor	
10.6 Laborator	• 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%
10.7 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 este OBLIGATORIE prezența la cel puțin 5 seminarii și 6 laboratoare. Studenții care nu au prezență la minimum 5 seminarii și 6 laboratoare nu se pot prezenta la examen nici în sesiunea de restanțe. • Pentru promovare sunt necesare următoarele criterii minimale: nota cel puțin 5 la lucrarea scrisă și nota finală cel puțin 5.			

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

Nu se aplică.

Data completării:

03.04.2025

Semnătura titularului de curs

Prof. dr. Gabriela CZIBULA

Semnătura titularului de seminar

Prof. dr. Gabriela CZIBULA

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".