

FIȘA DISCIPLINEI

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	Calculatoare si Tehnologia Informației
1.5 Ciclu de studii	Licență
1.6 Programul de studiu / Calificarea	Ingineria Informației

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare logică și funcțională						
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	3	2.5 Semestrul	5	2.6. Tipul de evaluare	C	2.7 Regimul disciplinei	Obligatorie DD

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	1 LP, 1 S
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					22
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					18
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					27
Tutoriat					11
Examinări					16
Alte activități:					-
3.7 Total ore studiu individual	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 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	Studentii vor participa la curs cu telefoanele mobile închise Studentii vor participa la curs cu laptop-urile închise; studentii 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

6. Competențele specifice acumulate

Competențe profesionale	<p>C1.1 Recunoașterea și descrierea conceptelor proprii calculabilității, complexității, paradigmelor de programare și modelării sistemelor de calcul</p> <p>C1.2 Utilizarea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea structurii și funcționării sistemelor</p> <p>C1.3 Construirea unor modele pentru diferite componente ale sistemelor de calcul</p> <p>C1.4 Evaluarea formală a caracteristicilor funcționale și nefuncționale ale sistemelor de calcul</p> <p>C1.5 Fundamentarea teoretică a caracteristicilor sistemelor proiectate</p>
Competențe transversale	<p>CT1 Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei</p> <p>CT3 Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> Să deprindă studentul cu paradigma programării declarative (programarea funcțională și programarea logică).
7.2 Obiectivele specifice	<ul style="list-style-type: none"> 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 urmării unor cursuri avansate de programare declarativă.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
<i>Programare Logică. Limbajul PROLOG</i>		
1. Programare și limbaje de programare. Programare imperativa vs. programare declarativă. Introducere. Recursivitate. Exemple.	<ul style="list-style-type: none"> 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.	<ul style="list-style-type: none"> 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).	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația 	

Directive de compilare. Expresii aritmetice și comparații. Operații de intrare / ieșire. Șiruri de caractere.	<ul style="list-style-type: none"> • 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 nedeterministe	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
5. Obiecte compuse și funcții. 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.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
6. Structuri de date recursive. Arborii ca structuri de date. Construirea și traversarea unui arbore. Arbori de căutare.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
8. Predicate de bază în Lisp. Predicate pentru liste; pentru numere. Funcții logice și aritmetice. Definierea funcțiilor utilizator. Ramificarea prelucrărilor. Metoda variabilei colectoare. Exemple.	<ul style="list-style-type: none"> • 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.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
10. Mecanisme definiționale evaluate Forma EVAL. Forme funcționale; funcțiile FUNCALL și APPLY. Expresii LAMBDA. Expresii LABEL. Exemple	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
11. Generatori, argumente funcționale. Funcții MAP. Forme iterative. Exemple.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
12. Alte elemente ale limbajului Lisp. Structuri de date. Macrodefiniții. Argumente opționale. Exemple.	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
13,14. Examen scris PROLOG+LISP	Lucrarea scrisă	
Bibliografie		
<ol style="list-style-type: none"> 1. CZIBULA G., POP H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastra, Cluj-Napoca, 2012 2. POP H.F., SERBAN G., Programare în Inteligența Artificială - Lisp și 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	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Studiu de caz 	
S2. Liste în Prolog	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Studiu de caz 	
S3. Prelucrarea listelor eterogene	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Studiu de caz 	
S4. Backtracking în Prolog	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Studiu de caz 	
S5. Prelucrarea listelor în LISP	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Studiu de caz 	
S6. Funcții MAP în LISP	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Studiu de caz 	
S7. Recapitulare	<ul style="list-style-type: none"> • 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. Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp. 3. Documentatia produselor: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog. 4. http://www.swi-prolog.org/		
8.3 Laborator	Metode de predare	Observații
		<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 2: Liste în Prolog	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 3: Gestiunea listelor (eterogene) în Prolog.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 4: Backtracking în Prolog	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 4: Proba practică Prolog	Lucrare practică	1 oră
Lab 5: Programare recursivă în Lisp	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 6: Programare recursivă în Lisp	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 7: Folosirea funcțiilor MAP.	<ul style="list-style-type: none"> • Lucrare de laborator • Explicația • Conversația • Modelarea 	
Lab 7: Proba practică Lisp	Lucrare practică	1 oră
Bibliografie		
<ol style="list-style-type: none"> 1. CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligența Artificială, Editura Albastra, Cluj-Napoca, 2012 2. Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp. 3. Documentatia produselor: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog. 4. 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	<ul style="list-style-type: none"> • Corectitudinea și completitudinea cunoștințelor asimilate. • Gradul de asimilare a limbajului de specialitate. 	Evaluare scrisă: examen scris Prolog+Lisp (curs 14)	60%
10.5 Seminar	<ul style="list-style-type: none"> • Activitatea din timpul seminariilor 	Evaluare a activității la seminarii	10%
10.6 Laborator	<ul style="list-style-type: none"> • 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)	10%
		Proba practică Lisp (1 oră, lab. 7)	10%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • 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 minim 5 seminarii și 6 laboratoare. • Pentru promovare sunt necesare următoarele criterii minimale: nota minim 5 la lucrarea scrisă și nota finală minim 5. 			

Data completării

27.04.2022

Semnătura titularului de curs

Prof. dr. Horia F. Pop

Semnătura titularului de seminar

Prof. dr. Horia F. Pop

Data avizării în departament

24.05.2022

Semnătura directorului de departament

Prof. dr. Laura Dioșan