

SYLLABUS / FIȘA DISCIPLINEI
1. Information on the study programme / Date despre programul de studii

1.1. Institution / Instituția de învățământ superior	Universitatea de Vest din Timișoara
1.2. Faculty / Facultatea	Matematică și Informatică
1.3. Department / Departamentul	Computer Science (Informatică)
1.4. Study program field	Computer Science (Informatică)
1.5. Study cycle/ Ciclul de studii	Bachelor / licență
1.6. Study programme / Programul de studii / calificarea*	Computer Science / Informatică în limba engleză / Database administration / <i>Administrator baze de date - 252101; Computer network administration / Administrator de rețea de calculatoare - 252301; Analyst / Analist - 251201; Research assistant in computer science / Asistent de cercetare în informatică - 214918; Teacher in secondary schools / Profesor în învățământul gimnazial - 233002; Programmer / Programator - 251202; Software systems designers / Proiectant sisteme informatice - 251101</i>

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Logic Programming						
2.2. Teacher in charge of the course / Titularul activităților de curs	Isabela Drămnesc						
2.3. Teacher in charge of the seminar / Titularul activităților de seminar	Isabela Drămnesc						
2.4. Study year / Anul de studii	2	2.5. Semester / Semestrul	1	2.6. Examination type / Tipul de evaluare: E(xam)/C(olloquim)	C	2.7. Course type / Regimul disciplinei: M(andatory)/ E(lective)/ F(acultative)	DO

3. Estimated study time (number of hours per semester) /Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Attendance hours per week / Număr de ore pe săptămână	3	out of which din care: 3.2 lecture/ curs	2	3.3. seminar/laborator	1
3.4. Attendance hours per semester / Total ore din planul de învățământ	42	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	14
Distribution of the allocated amount of time / Distribuția fondului de timp*					hours/ore
Individual study /Studiu după manual, suport de curs, bibliografie și notițe					28
Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					13
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					14
Exams / Examinări					4

Tutoring / Tutorat		7
3.7. Total number of hours of individual study / Total ore studiu individual	66	
3.8. Total number of hours per semester / Total ore pe semestru	108	
3.9. Number of credits (ECTS) / Număr de credite	4	

4. Prerequisites (if it is the case) / Precondiții (acolo unde e cazul)

4.1. curriculum / de curriculum	Algorithms, logic for computer science / Algoritmi, logica
4.2. skills / de competențe	The ability of problem modeling, programming. / Abilitatea de a modela probleme, programare.

5. Requirements (if it is the case) / Condiții (acolo unde e cazul)

5.1. for the lecture / de desfășurare a cursului	Room with whiteboard and videoprojector / Sala de curs cu tabla și videoprojector
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	Room with computers with SWI-Prolog installed / Sala de laborator cu calculatoare cu SWI-Prolog instalat

6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	<ul style="list-style-type: none"> Solving problems in logic programming paradigm requires modeling skills development and representation issues in a declarative manner, reasoning and abstract thinking skills. / Rezolvarea problemelor în paradigma de programare logică presupune dezvoltarea de abilități de modelare și reprezentare a problemelor într-o manieră declarativă, abilități de raționament și gândire abstractă. The ability to solve Prolog programming problems encountered so far (ex. Sort of lists of tens / hundreds of thousands of elements, automatic demonstration of theorems based on resolution, the implementation of automatic natural language processing elements) / Abilitatea de a rezolva în Prolog problemele de programare întâlnite până acum (ex. Sortări ale unor liste cu zeci/sute de mii de elemente, demonstratoare automate de teoreme bazate pe rezoluție, implementarea de automate,
---	--

	<p>elemente de prelucrare a limbajului natural),</p> <ul style="list-style-type: none"> • The ability of writing efficient logic programs / Abilitatea de a scrie programe logice eficiente, • The ability of understanding the theoretical foundations of logic programming / Abilitatea de a înțelege fundamentele teoretice ale programării logice.
Transversal skills / Competențe transversale	<ul style="list-style-type: none"> • Skills for better problem modeling (necessary to express the problems as logic programs) / Abilități mult mai bune de modelare (necesare pentru exprimarea problemelor ca programe logice), • Develop comprehensive Prolog applications that can interact with other programming languages or applications / Dezvoltarea de aplicații Prolog complete capabile să interacționeze cu alte limbaje de programare sau aplicații, • Gaining programming skills useful in other disciplines related to Artificial Intelligence / Dobândirea de competențe de programare utile și în alte discipline înrudite cu Inteligența Artificială.

7. Objectives of the course / Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1. General objective / Obiectivul general al disciplinei	<p>Gaining fundamental knowledge in logic programming. Understanding the theoretical bases of logic programming. Acquiring and applying advanced programming techniques to improve the efficiency of logic programs. / Insusirea de cunoștințe fundamentale în programarea logică. Înțelegerea bazelor teoretice ale programării logice. Insușirea și aplicarea tehnicilor avansate de programare în vederea îmbunătățirii eficienței programelor logice.</p>
7.2. Specific objectives / Obiectivele specifice	<ul style="list-style-type: none"> • The ability to formalize the knowledge about a problem in a concise and clear logically / Abilitatea de a formaliza cunoștințele despre o problemă într-un mod concis și clar din punct de vedere logic, • Practicing a sequential way of thinking based on logical deductions (if something happens, then something happens) and to identify the sequence of steps that lead to finding the desired solution / Practicarea unui mod de gândire secvențial, bazat pe deducții logice (<i>dacă</i> ceva se întâmplă, <i>atunci</i> altceva se întâmplă) și de a identifica acele secvențe de pași care conduc la găsirea soluției dorite,

	<ul style="list-style-type: none"> • The ability to solve problems effectively Prolog presented in the lectures and laboratories / Abilitatea de a rezolva eficient în Prolog probleme de tipul celor prezentate la curs și laborator, • Deepening the theoretical bases of logic programming and identifying where this style of programming is convenient / Aprofundarea bazelor teoretice ale programării logice și identificarea situațiilor în care acest stil de programare este convenabil, • Awareness of limits as Prolog programming language / Conștientizarea limitărilor Prolog ca limbaj de programare, • The ability to develop complete applications in Prolog / Abilitatea de a dezvolta aplicații complete în Prolog.
--	---

8. Content / Conținuturi*

8.1. Lecture / Curs	Teaching strategies / Metode de predare	Remarks, details / Observații
1) Motivation. Introduction. A short history of Prolog. Facts, rules, queries. Syntax. Constants. Variables. Structures. Operators. Equality and unification. / Motivație. Introducere. Istoric Prolog. Fapte, reguli, întrebări. Sintaxă. Constante. Variabile. Structuri. Operatori. Egalitate și unificare.	Lecture, conversation, illustration / Prelegere, conversație, exemplificare	References/Referinte: 1. Isabela Drămnesc – lecture slides http://web.info.uvt.ro/~idramnesc 2. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994
2) Data structures. Lists. Comparisons, recursive mapping. / Structuri de date. Liste. Comparări, mapare recursiva.	Idem	References/Referinte: 1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc 2. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994
3) Recursion. Difference lists / Recursivitate. Liste cu diferențe.	Idem	References/Referinte: 1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc 2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog,

		<p>2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000</p> <p>3. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994</p>
4) Accumulators / Acumulatori.	Idem	<p>References/Referinte:</p> <p>1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc</p> <p>2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog, 2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000</p> <p>3. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994</p>
5) Backtracking and the cut predicate. Common uses of the cut predicate. Fail / Backtracking și predicatul de tăiere. Exemple de utilizare. Negația ca insucces	Idem	<p>References/Referinte:</p> <p>1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc</p> <p>2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog, 2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000</p> <p>3. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994</p>
6) Efficiency in Prolog. Coding style. / Prolog eficient. Considerații asupra stilului de programare.	Idem	<p>References/Referinte:</p> <p>1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc</p> <p>2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog, 2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000</p> <p>3. M. A. Covington, Efficient Prolog:</p>

		A Practical Guide, Research Report AI-1989-08, The University of Georgia, Athens, Georgia, 1989.
7) I/O in Prolog. Defining new operators / I/O în Prolog. Lucrul cu fișiere. Definirea de noi operatori.	Idem	References/Referinte: 1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc 2. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994
8) First order predicate logic. The clausal form. Unification / Logica predicatelor de ordinul I. Forma clauzală. Unificare.	Idem	References/Referinte: 1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc 2. Robert Kowalski, Logic for Problem Solving, North Holland New York, Amsterdam, Oxford, 1979. 3. M. Ben-Ari, Mathematical Logic for Computer Science, Springer Verlag, London, 2nd edition, 2001.
9) Horn clauses. Resolution / Clauze Horn. Principiul rezoluției.	Idem	References/Referinte: 1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc 2. Robert Kowalski, Logic for Problem Solving, North Holland New York, Amsterdam, Oxford, 1979. 3. M. Ben-Ari, Mathematical Logic for Computer Science, Springer Verlag, London, 2nd edition, 2001.
10) SLD Resolution, soundness, completeness. / Rezoluție SLD corectitudine și completitudine.	Idem	References/Referinte: 1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc 2. Robert Kowalski, Logic for

		<p>Problem Solving, North Holland New York, Amsterdam, Oxford, 1979.</p> <p>3. M. Ben-Ari, Mathematical Logic for Computer Science, Springer Verlag, London, 2nd edition, 2001.</p>
11) Constraint Logic Programming / Programare logică cu constrângeri.	Idem	<p>References/Referinte:</p> <p>1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc</p> <p>2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog, 2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000</p> <p>3. M. Ben-Ari, Mathematical Logic for Computer Science, Springer Verlag, London, 2nd edition, 2001</p>
12) Concurrent Logic Programming / Programare logica concurenta.	Idem	<p>References/Referinte:</p> <p>1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc</p> <p>2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog, 2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000</p> <p>3. M. Ben-Ari, Mathematical Logic for Computer Science, Springer Verlag, London, 2nd edition, 2001</p>
13) Revision / Recapitulare	Idem	
14) Colloque / Colocviu		
<p>Recommended bibliography / Bibliografie</p> <p>1. Isabela Drămnesc – slides http://web.info.uvt.ro/~idramnesc</p> <p>2. Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog, 2nd Edition, copyright Ulf Nilsson and Jan Maluszynski, 2000.</p>		

3. Paul Brna, Prolog Programming A First Course, Copyright Paul Brna, 1988.
4. M. A. Covington, Efficient Prolog: A Practical Guide, Research Report AI-1989-08, The University of Georgia, Athens, Georgia, 1989.
5. M.A. Covington, D. Nute, A. Vellino, Prolog Programming in Depth Prentice Hall, New Jersey, 1997.
6. Robert Kowalski, Logic for Problem Solving, North Holland New York, Amsterdam, Oxford, 1979.
7. C.S. Mellish, W. F. Clocksin, Programming in Prolog, Springer Verlag Telos, 4th edition, 1994.
8. Jan Wielemaker, SWI-Prolog 5.6.60 Reference Manual, ©1990-2008 , University of Amsterdam. <http://gollem.science.uva.nl/SWI-Prolog/Manual>.
9. M. Ben-Ari, Mathematical Logic for Computer Science, Springer Verlag, London, 2nd edition, 2001

8.2. Seminar, lab / Seminar, laborator	Teaching/learning strategies / Metode de predare/ învățare	Remarks, details / Observații
1) Introduction in SWI-Prolog. Facts, rules, queries, constants, variables, unification. / Introducere in SWI-Prolog. Fapte, reguli, interogari, constante, variabile, unificare.	Questioning, dialogue, collaborative learning / Problematizare, dialog, învățare prin colaborare	The students can access directly online the requests for the exercises (http://web.info.uvt.ro/~idramnesc). The teacher gives details/explains, answers to the students questions and checks/evaluates the way how the students have solved the exercises./ Studentii au acces la sinteza aferentă tematicii de laborator și la enunțurile problemelor recomandate spre rezolvare (http://web.info.uvt.ro/~idramnesc). Cadrul didactic oferă detalii suplimentare, răspunde întrebărilor studenților și verifică/ evaluează modul în care studenții au rezolvat problemele
2) Data structures, lists, recursion / Structuri de date, liste, recursivitate.	Idem	Idem
3) Accumulators, backtracking, the cut predicate / Acumulatori, backtracking, predicatul cut.	Idem	Idem
4) I/O Operations with files / I/O Operatii cu fisiere	Idem	Idem
5) Natural language. Efficiency in Prolog. Strings / Limbaj natural. Eficienta in Prolog. Stringuri	Idem	Idem

6) Revision / Recapitulare	Idem	Idem
7) Examination / Examen (proba practica)		Written and practical test. / Test scris si practic.
Recommended bibliography / Bibliografie		
The same as above/Idem cu cea de mai sus.		

9. Correlations between the content of the course and the requirements of the IT field / Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

- The content is consistent with the structure of similar courses from other universities and covers the fundamental aspects necessary in order to become familiar with issues of designing and analysis of algorithms. The ability to identify, design, implement and analyze algorithms is essential for any activity in the field of computer science. The course develops skills that are useful beyond the logic programming paradigm. The skills offered by this discipline are needed for an IT specialist to identify effective solutions for solving concrete problems, regardless of their industry. / Conținutul este în concordanță cu structura cursurilor similare de la alte universități și acoperă aspectele fundamentale necesare familiarizării cu problematica proiectării și analizei algoritmilor. Abilitatea de a identifica, proiecta, implementa și analiza algoritmi este esențială pentru orice activitate din domeniul informaticii. Cursul dezvoltă deprinderi de modelare, care sunt utile dincolo de paradigma de programare logică. Competențele oferite de această disciplină sunt necesare unui specialist IT pentru a identifica soluții eficiente de rezolvare a unor probleme concrete, indiferent de domeniul specific de activitate.

10. Evaluation / Evaluare*

Activity / Tip de activitate	10.1. Evaluation criteria / Criterii de evaluare**	10.2. Evaluation methods / Metode de evaluare***	10.3. Weight in the averaged mark / Pondere din nota finală
10.4. Lecture / Curs	1) Learning the concepts presented in the lectures / Invatarea conceptelor prezentate la cursuri 2) The application of the concepts in concrete cases / Aplicarea conceptelor in cazuri concrete	Colloquy (Written test during the last lecture) / Colocviu (Test scris in timpul ultimului curs)	50%
10.5. Seminar/ lab	Exercises related to the lectures / Exerciții referitoare la conținutul cursurilor predate	Homework + activity / teme + activitate	25 %
		Written and practical test during the last laboratory / test scris si practic in timpul	25%

		ultimului laborator	
10.6. Minimal knowledge for passing / Standard minim de performanță			
<p>The minimal knowledge for passing (for grade 5)/ Standard minim (cunoștințe și aptitudini necesare pentru nota 5)</p> <ul style="list-style-type: none"> • The ability to design and implement correctly simple algorithms in Prolog / capacitatea de a implementa corect algoritmi simpli în Prolog. • The ability of describing a simple recursive predicate in Prolog (definition + call) / capacitatea de a descrie un predicat recursiv simplu (definiție + apel); • The ability of transforming a recursive predicate into a predicate which uses accumulators / capacitatea de a transforma un predicat recursiv într-un predicat cu acumulator(i); <p>The final grade is calculated as a weighted average of the grades given for the components specified in 10.4 and 10.5. The exam is passed if the average is at least 5 (not necessary as each grade to be greater than 5). At each exam sessions the grade is calculated by the same rule.</p> <p>Remark: The students may attend office hours (one hour / week according to the schedule set out at the beginning of the semester) where the teacher of the course and / or laboratory answers to the students questions and provides further explanations related to the course content, and to the applications from laboratories.</p> <p>/ Nota finală se calculează ca medie ponderată a notelor acordate pentru componentele specificate la 10.4 și 10.5. Examenul se consideră promovat dacă media este cel puțin 5 (nu e necesar ca fiecare notă să fie mai mare de 5) . La fiecare dintre sesiunile de examen (inclusiv cele de restanță și măriri) nota se calculează după aceeași regulă.</p> <p>Obs: Studenții pot participa la orele de consultații (1 ora/săptămână conform planificării stabilite la începutul semestrului) în cadrul cărora titularul de curs și/sau seminar/laborator răspunde întrebărilor studenților și oferă explicații suplimentare legate de conținutul cursului, aplicațiile de la laborator și teme.</p>			

Date/ Data completării

1.10.2016

Signature (lecture) /
Semnătura titularului de curs

Signature (seminar)
Semnătura titularului de seminar

Signature (director of the department)
Semnătura directorului de departament
Conf.dr. Victoria Iordan