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	Master / master
1.6. Study programme / Programul de studii / calificarea*	Artificial Intelligence and Distributed Computing

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Special Topics in Artificial Intelligence						
2.2. Teacher in charge of the course / Titularul activităților de curs	Madalina Erascu						
2.3. Teacher in charge of the seminar / Titularul activităților de seminar	Madalina Erascu						
2.4. Study year / Anul de studii	2	2.5. Semester / Semestrul	1	2.6. Examination type / Tipul de evaluare: /C(olloquim)	C	2.7. Course type / Regimul disciplinei: E(lective)	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	30
Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate	10
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	20
Exams / Examinări	2
Tutoring / Tutorat	2

3.7. Total number of hours of individual study / Total ore studiu individual	130
3.8. Total number of hours per semester / Total ore pe semestru	60

3.9. Number of credits (ECTS) / Număr de credite	5

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

4.1. curriculum / de curriculum	Computational logic, algorithmics
4.2. skills / de competențe	Mathematical knowledge and problem-solving skills

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

5.1. for the lecture / de desfășurare a cursului	Variant face-to-face: room with video projector; personal computer with internet connection Variant online: personal computer with internet connection
5.2. for the seminar, laboratory / de desfășurare a seminarului/laboratorului	Same as above

6. Acquired skills / Competențe specifice acumulate

Professional skills / Competențe profesionale	Presentation and understanding of (1) the importance of logical theories in the verification of programs, with (2) application to use-cases using emerging technologies (IoT, AI).
Transversal skills / Competențe transversale	The ability of communicating knowledge about the usage of logical methods for different problems (optimization, verification of certain program properties, etc.)

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

7.1. General objective / Obiectivul general al disciplinei	Understanding how logical methods are useful for verifying different properties of software.
7.2. Specific objectives / Obiectivele specifice	*Knowledge objectives:* understanding and usage propositional and predicate logic in program analysis and verification (optimization, loop invariants, termination terms, program specification). *Abilitation objectives*: basic and advanced usage of dedicated software SAT solvers, SMT solvers, Mathematica. *Atitudinal objectives*: motivation and argumentation of the importance of logical methods in modelling, verifying and optimizing the software.

8. Content / Conținuturi*

8.1. Lecture / Curs	**Teaching strategies /**	**Remarks, details / Observații**

	Metode de predare	
C1-2 (4h). *Organizational matters and course motivation.*	Lecture, conversation, illustration	References: • M. Erascu slides
C3-C4 (*4h*). *Propositional logic. SAT solving. DPLL and CDCL algorithms*	Same as above	Same as above
C5-6 (*2h*). *SAT@Work: Problem formalization with propositional logic; DIMACS format. SAT solvers*	Same as above	Same as above
C7 (*2h*). *First-Order Theories. Decidability*	Same as above	Same as above
C8 (*2h*). *Principles of SMT solving. SMT-LIB format*	Same as above	Same as above
C9 (*2h*). *Linear real arithmetic. Fourier Motzkin algorithm. Examples.*	Same as above	Same as above
C10 (*2h*). *Linear integer arithmetic: Branch and bound algorithm, Examples.*	Same as above	Same as above
C11-12 (4h). SMT@Work: Problem formalization with SMT; SMT solvers	Same as above	Same as above
C13-14 (4h). Examination (Group projects)	Same as above	Same as above

Recommended bibliography / Bibliografie

[1] C.-L. Chang, R. C. T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Computer Science Classics
[4] L. de Moura, N. Bjorner. *Satisfiability Modulo Theories: Introduction and Applications.*
[5] J. Woodcock et al. *Formal Methods: Practice and Experience*
[6] Formal Verification of Object-Oriented Software: http://www.cost-ic0701.org/
[7] A. Biere, M. Heule, H. Van Maaren, T. Walsh. *Handbook of Satisfability*. IOS Press 2009
[8] A. Bradley, Z. Manna. The Calculus of Computation. Decision procedures with Applications to Verification. Springer 2007
[9] D. Kroening, O. Strichman. *Decision Procedures An Algorithmic Point of View*. Springer 2008
[10] J.-C. Régin and M. Rezgui. *Discussion about Constraint Programming Bin Packing Models*. AI for Data Center Management and Cloud Computing: Papers from the 2011 AAAI Workshop (WS-11-08)
[11] Edited by Francesca Rossi, Peter van Beek, Toby Walsh. Handbook of Constraint Programming, 2006, Elsevier. In particular Chapter 10.
[12] E. Abraham. SAT-checking. Lecture Notes, RWTH Aachen, https://ths.rwth-aachen.de/teaching/ws14/lecture-sat-checking/

8.2. Seminar, lab / Seminar, laborator	**Teaching/learning strategies / Metode de predare/ invățare**	**Remarks, details / Observații**

L1-7 (*2h*). *Exercises on the topics presented as well as formalization of problems and application of SAT/SMT solvers for different practical problems.*	Questioning, dialogue, collaborative learning	Each lab will be available online. The students will have time until the next to solve it. At the next lab meeting they will present their work and receive feedback.
Recommended bibliography / Bibliografie - same as for the lecture - SAT solvers - Z3 SMT solver (https://github.com/Z3Prover/z3), OptiMathSAT (http://optimathsat.disi.unitn.it)		

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 of the lecture is similar to others, on the same topic, from other universities. It covers the fundamental notions for understanding why formal methods for software development are so important. Currently, the lecture seems to be not that useful for ordinary IT companies in Romania. However, formal methods are necessary for safety-critical systems (avionics, cars, medical devices) becoming mandatory. We foresee a need of them in the next decade in Romania, too.

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	• Group project (3-4 students)	Project presentation+report. Based on the topics presented in the lecture, or related to them, a list of projects will be available on the first week of classes. **Students can also propose topics but they must be agreed with the teacher.**	• 20% for the draft version (deadline around Week 7); a report must be uploaded on Classroom, • 50% (25% presentation+25%report) for the final version
10.5. Laboratory	• The ability to reproduce research results with instructor advice/help • The ability to deliver the reproduced results in a presentation and demo	Each lab will be available online. The students have approx. 1 week to solve it. At the next lab meeting they will present their work and receive feedback.	30%
10.6. Minimal knowledge for passing / Standard minim de performanță			
Minimal knowledge for passing (grade 5): acquiring fundamental understanding of the knowledge of			

propositional logic, first order logic, program analysis and verification.
The final grade is computed as a weighted average of the grades given for the components specified in 10.4- 10.5. The exam is passed if the average is equal or greater than 4.1 (each component grade from 10.4-10.5 must be greater than 4.1). If the final grade is greater than equal to 4.1 means 5, greater than equal to 5.1 means 6, …, greater than equal to 9.1 means 10.
Minimal requirements for passing the laboratory: 2 presentations of the lab work during the lab.
Minimal requirements for passing the course: submission of both draft and final versions of the project and both should be grades with at least 5.
There is no mandatory presence requirement.

Date/ Data completării	Signature (lecture) / Semnătura titularului de curs	Signature (seminar) Semnătura titularului de seminar
13.09.2021	Madalina Erascu	Madalina Erascu

Signature (director of the department)
Semnătura directorului de departament