SYLLABUS / FIŞA DISCIPLINEI

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

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

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Formal Methods in Software Development
2.2. Teacher in charge of the course / Titularul activităţilor de curs	Madalina Erascu
2.3. Teacher in charge of the seminar / Titularul activităţilor de seminar	Madalina Erascu

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)	E

3. Estimated study time (number of hours per semester) /Timpul total estimat (ore pe semestru al activităţilor 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 învăţământ	42	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	14

Distribution of the allocated amount of time / Distribuţia fondului de timp*	**hours / ore**
Individual study /Studiu după manual, suport de curs, bibliografie şi notiţe	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 şi 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) / Precondiţii (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) / Condiţii (acolo unde e cazul)

5.1. for the lecture / de desfăşurare a cursului	Classroom with blackboard and video projector
5.2. for the seminar, laboratory / de desfăşurare a seminarului/laboratorului	Classroom with blackboard, video projector and computers. We will install a virtual machine with the needed software.

6. Acquired skills / Competenţe specifice acumulate

Professional skills / Competențe profesionale	Presentation and understanding of (1) the importance of logical theories in the verification of programs, (2) static analysis of programs.
Transversal skills / Competențe transversale	The ability of communicating knowledge about different notions from formal methods of software.

7. Objectives of the course / Obiectivele disciplinei (reieşind din grila competenţelor specifice acumulate)

7.1. General objective / Obiectivul general al disciplinei	Understanding of different notions from formal methods of software (static program analysis).
7.2. Specfic objectives / Obiectivele specifice	*Knowledge objectives:* understanding and usage propositional and predicate logic in program analysis and verification (loop invariants, termination terms, program specification).

	Abilitation objectives: basic usage of dedicated software RISC Proof Navigator, Dafny, Mathematica. *Atitudinal objectives*: motivation and argumentation of the importance of formal verification of software.

8. Content / Conţinuturi*

8.1. Lecture / Curs	**Teaching strategies / Metode de predare**	**Remarks, details / Observaţii**
C0. *Organizational matters and course motivation*		
C1 (*2h*). *Propositional Logic (refresher)*	Lecture, conversation, illustration	References: ● M. Erascu slides ● specified at the beginning of each lecture
C2 (*2h*). *First-Order Logic (refresher)*	Same as above	Same as above
C3 (*2h*). *Solving homework from Week 1&2*	Same as above	Same as above
C4 (*2h*). *Proof Techniques*	Same as above	Same as above
C5 (*2h*). *Herbrand's Theorem*	Same as above	Same as above
C6 (*2h*). *Resolution Principle*	Same as above	Same as above
C7 (*2h*). *Solving homework from Week 5&6*	Same as above	Same as above
C8 (*2h*). *Midterm 1*	Same as above	Same as above
C9 (*2h*). *Reasoning about Programs I (Hoare logic, predicate transformers)*	Same as above	Same as above
C10 (*2h*). *Reasoning about Programs II (Dafny)*	Same as above	Same as above
C11 (*2h*). *SMT Solving*	Same as above	Same as above
C12 (*2h*). *Midterm 2*	Same as above	Same as above
C13 (*2h*). *Quantifier Elimination*	Same as above	Same as above

C14 (*2h*). *Program Synthesis*	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

[2] C.A.R. Hoare. *An axiomatic Basis for Computer Programming.*

[3] M. Huth, M. Ryan. *Logic in Computer Science. Modelling and Reasoning about Systems*.

[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 Satisfiability*. 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

8.2. Seminar, lab / Seminar, laborator	**Teaching/learning strategies / Metode de predare/ invăţare**	**Remarks, details / Observaţii**
L1 (*2h*). *SAT Solvers*	Questioning, dialogue, collaborative learning	The labs will be available on the website before the lecture and students have to work on those subjects during the lab.
L2 (*2h*). *Basic Proof Techniques*	Same as above	Same as above
L3 (*2h*). *Reasoning about Programs I*	Same as above	Same as above
L4 (*2h*). *Reasoning about Programs I*	Same as above	Same as above
L5 (*2h*). *Quantifier Elimination*	Same as above	Same as above
L6 (*2h*). *Robust software development with mbeddr*	Same as above	Same as above
L7 (*2h*). *C-level verification with CBMC*	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

[2] A. Bradley, Z. Manna. The Calculus of Computation. Decision procedures with Applications to Verification. Springer 2007

[3] D. Kroening, O. Strichman. *Decision Procedures An Algorithmic Point of View*. Springer 2008

[4] http://mbeddr.com/

[5] www.wolfram.com

[6] http://rise4fun.com/dafny

9. Correlations between the content of the course and the requirements of the IT field / Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi 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	Knowledge and application of notions from C1 - C7.	Midterm I	20%
	Knowledge and application of notions from C9 - C11.	Midterm II	10%
	Knowledge and application of notions from C1 - C14.	Written exam in the exam session	40%
10.5. Seminar/ lab	The ability to learn and apply concepts presented during the lectures.	Homeworks and activity (oral examination)	20%
10.6. Projects	The ability to implement an algorithm in a certain programing language for certain notions presented during the lecture as well as its presentation oral and written form.	homeworks and projects	20%
10.8 Bonuses	The ability to solve problems on-site during the lecture/lab, the homeworks in the lectures from Week 3&7, as well as other problems announced during the semester	oral/written questions and homeworks	10%

10.6. Minimal knowledge for passing / Standard minim de performanţă

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- 10.8, however it does not have negative impact on the grade if this activity is not fulfilled. The exam is passed if the average is equal or greater than 4.1 (not necessary as each note to be greater than 4.1). The start at Midterms and Final Exam is 0. 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.

At each exam sessions (including reexamination and improvements), the score is computed by the same rule. Midterm I-II can be retaken a single time and that is in the first exam session, but in a time framework of 90 minutes.

There is no mandatory presence requirement, however, note that your lab grade is based on your activity during the semester (you will be graded at the end of each lab).

Note: Students may attend office hours (2 modules / week according to the schedule set out at the beginning of the semester) where the lecturer (course/seminar) answers questions students and provides further explanations related to course content, applications from seminary themes.

Date/ Data completării	Signature (lecture) / Semnătura titularului de curs	Signature (seminar) Semnătura titularului de seminar
21.09.2017	Madalina Erascu	Madalina Erascu

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