SYLLABUS

1. Data about the program of study

1.1	Institution	The Technical University of Cluj-Napoca
1.2	Faculty	Faculty of Automation and Computer Science
1.3	Department	Computer Science
1.4	Field of study	Computer Science and Information Technology
1.5	Cycle of study	Master of Science
1.6	Program of study/Qualification	Artificial Intelligence and Vision
1.7	Form of education	Full time
1.8	Subject code	7.2

2. Data about the subject

2.1	Subject name			***Languages and Type Systems***			
2.2	Subject area			Computer Science and Information Technology			
2.2	Course responsible/lecturer			Prof.dr.eng. Eneia Nicolae Todoran – Eneia.Todoran@cs.utcluj.ro			
2.3	Teachers in charge of seminars			Prof.dr.eng. Eneia Nicolae Todoran – Eneia.Todoran@cs.utcluj.ro			
2.4 Year of study	I	2.5 Semester	2	2.6 Assessment	E–**e**xam, C–**c**olloq., V-**v**erif.		E
2.7 Subject category	Formative category: DA – advanced, DS – speciality, DC – complementary						DS
	Optionality: DI – imposed, DO – optional (alternative), DF – optional (free choice)						DO

3. Estimated total time

3.1 Number of hours per week	3	of which	3.2 Course	2	3.3 Seminar	1	3.3 Laborator	-	3.3 Proiect	-
3.4 Total hours in the curriculum	42	of which	3.5 Course	28	3.6 Seminar	14	3.6 Laborator	-	3.6 Proiect	-

3.7 Individual study:	
(a) Manual, lecture material and notes, bibliography	20
(b) Supplementary study in the library, online and in the field	10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	15
(d) Tutoring	10
(e) Exams and tests	3
(f) Other activities	-

3.8 Total hours of individual study (sum (3.7(a)...3.7(f)))	58
3.9 Total hours per semester (3.4+3.8)	100
3.10 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	Programming Languages (undergraduate course)
4.2	Competence	Scientific, engineering and mathematical foundations

5. Requirements (where appropriate)

5.1	For the course	Interactive course, for the maximum grade a minimum of 70% course attendance is required
5.2	For applications	For the maximum grade the student should attend at least 70% of the seminar hours

6. Specific competences

Professional competences	C1 - Working with mathematical methods and models, advanced engineering and IT techniques and technologies • C1.1 - Presentation of advanced theoretical and practical concepts and principles related to intelligent systems and artificial vision • C1.2 - The use of specific theories and tools (algorithms, schemes, models, protocols, etc.) to explain the structure and operation mode of the latest intelligent and artificial vision systems reported in the specialized scientific literature • C1.3 - Use of models for different components of complex artificial vision and intelligent systems under partial specification conditions • C1.4 - Formal and comparative evaluation of the characteristics of intelligent and complex artificial vision systems • C1.5 - Theoretical substantiation of the characteristics of complex intelligent and artificial vision systems, based on modern theoretical and practical trends C2 - Use of computing techniques in the fields of artificial intelligence and artificial vision and their applications • C2.1 - Identification and description of structures and operation mode for intelligent and artificial vision components and systems • C2.2 - Explaining the role, interactions and functional characteristics of the components of intelligent and artificial vision systems reported in the specialized scientific literature • C2.3 - Development of original software and hardware components, of intelligent and artificial vision systems, using algorithms, design methods, protocols, programming languages, data structures, technologies • C2.4 - Evaluation of functional and non-functional characteristics of intelligent and artificial vision systems, based on specific metrics • C2.5 - Implementation of intelligent and artificial vision systems
Cross competences	N/A

7. Discipline objectives (as results from the *key competences gained*)

7.1	General objective	The main objective of this discipline is to provide specific knowledge and to prepare students for the use of formal and semantic models in the design and verification of computing systems. The knowledge is presented in the context of programming and specification languages and process calculi, with an emphasis on static type checking, dynamic semantics, performance modelling and formal verification of the properties of computing systems.
7.2	Specific objectives	To achieve these general objectives, students will: • Learn to specify and formally design computer languages and systems via their operational semantics and type system • Learn to formally verify properties of languages and systems • State and prove properties of programs based on their semantics • Learn techniques for designing and verifying the properties of languages and systems (induction, fixed point semantics, bisimulation) • Learn to apply advanced design principles and paradigms • Study how semantic techniques allow solving complex problems in formal design, performance evaluation, formal verification • Learn to apply formal methods in the specification, development and verification of software systems

8. Contents

8.1. Lecture (syllabus)	Number of hours	Teaching methods	Notes
Introduction and overview of the course	2	Interactive course, exposition, examples, questions, discussions	
Semantic styles, operational semantics	2		
Untyped Lambda Calculus (ULC): syntax, evaluation relation	2		
ULC: Nameless representation of terms (De Bruijn terms)	2		
Simply Typed Lambda Calculus (STLC): syntax and typing relation	2		
STLC: properties of typing (progress and preservation theorems)	2		
Simple extensions: records, variants, general recursion, lists	2		
Semantic interpreter for STLC, Haskell implementation	2		
Introduction to Dependent Types (DT)	2		
Introduction to process algebras and bisimulation	2		
Bisimulation and algebraic semantics: concurrency and nondeterminism	2		
Introduction to CCS (Calculus of Communicating Systems)	2		
Bisimulation and algebraic semantics in CCS (1)	2		
Bisimulation and algebraic semantics in CCS (2)	2		

Bibliography

1. B. Pierce, Types and Programming Languages, MIT Press, 2002.
2. B. Pierce, Advanced Topics in Types and Programming Languages, MIT Press, 2005.
3. B. Pierce et al, Software Foundations, https://softwarefoundations.cis.upenn.edu, 2023.
4. R. Milner. Communicating and mobile systems: the pi-calculus, Cambridge Univ. Press, 1999.
5. D. Sangiorgi, Introduction to Bisimulation and Coinduction, Cambridge University Press, 2011.
6. R. Harper, Practical Foundations for Programming Languages, Cambridge University Press, 2016.
7. Types and Programming Languages (lecture notes, adapted from 1), Technical University of Cluj-Napoca, https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2020.pdf, 2020.
8. E.N. Todoran. Introducere in Semantica Limbajelor de Programare, Note de curs si seminar, Universitatea Tehnica din Cluj-Napoca, http://users.utcluj.ro/~eneia/aplc-2016.pdf, 2016.

8.2. Seminars/ Laboratory / Project	Number of hours	Teaching methods	Notes
Structured operational semantics	2		
Semantic design with transition systems	2		
Design and Haskell implementation of a semantic interpreter for STLC	2		
Functional programming with DT	2		
Algebraic laws for CCS in bisimulation semantics (1)	2		
Algebraic laws for CCS in bisimulation semantics (2)	2		

Bibliography

1. B. Pierce, Types and Programming Languages, MIT Press, 2002.
2. B. Pierce et al, Software Foundations, https://softwarefoundations.cis.upenn.edu, 2023.
3. E.M. Clarke et al, Handbook of Model Checking, Springer, 2018.
4. D. Sangiorgi, Introduction to Bisimulation and Coinduction, Cambridge University Press, 2011.
5. Types and Programing Languages (lecture notes, adapted from 1), Tehnical University of Cluj-Napoca, https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2020.pdf, 2020.
6. PRISM probabilistic model checker www.prismmodelchecker.org

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

This course introduces basic knowledge in the fields of Semantics, Type Systems and Formal Methods. The presentation is made in the context of programming and specifications languages and process calculi. The languages and systems under consideration are described mathematically using formal syntax and are endowed with formal (static and dynamic) semantics. From an engineering perspective, this discipline offers important knowledge for the development of computing systems that impose strict quality standards: reliability, operational safety, measurable performance, etc. Each student must elaborate a scientific paper (an essay or a technical report). For the elaboration of the paper students can choose from a wide variety of topics: advanced topics in types and programming languages, model checking, dependent types, session types, runtime verification, nature inspired models of computation (DNA computing, membrane computing), process calculi, etc. The content of the discipline is synchronized with the latest advances in the field, based on monographs, studies and courses taught at prestigious universities in Europe and the USA.

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
10.4 Course	The ability to solve problems specific to the field, attendance and activity during the course hours	Final exam	75%
10.5 Seminars /Laboratory/Project	The ability to solve problems specific to the field, attendance and activity in seminar classes	Elaboration of a scientific paper Seminar assignments	25%
10.6 Minimum standard of performance			
Modeling and solving semantic design problems, by using the formal apparatus specific to the field (type systems, operational semantics, bisimulation)			

Date of filling in:		**Title Surname Name**	**Signature**
	Lecturer	Prof.dr.eng. Eneia Nicolae Todoran	
	Teachers in charge of application	Prof.dr.eng. Eneia Nicolae Todoran	

Date of approval in the department 20.02.2024	Head of department Prof.dr.ing. Rodica Potolea
Date of approval in the faculty council 22.02.2024	Dean Prof.dr.ing. Mihaela Dinsoreanu