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	Masters
1.6 Program of study / Qualification	Data Science / Masters
1.7 Form of education	Full time

2. Data about the subject

2.1 Subject name	***Advanced Topics in Functional Programming***			Subject code	**7.3**
2.2 Course responsible / lecturer	Assoc. prof. dr. eng. Slăvescu - Radu.Razvan.Slavescu@cs.utcluj.ro				
2.3 Teachers in charge of seminars / Laboratory / project	Assoc. prof. dr. eng. Slăvescu - Radu.Razvan.Slavescu@cs.utcluj.ro				
2.4 Year of study	I	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V – verification)	E
2.7 Subject category	Formative category: DF – fundamental, DS – speciality, DC – complementary				DF
	Optionality: DOB – mandatory, DOP – optional (alternative), DFA – optional (free choice)				DOP

3. Estimated total time

3.1 Number of hours per week	3	of which:	Course	2	Seminars	-	Laboratory	1	Project	-
3.2 Number of hours per semester	42	of which:	Course	28	Seminars	-	Laboratory	14	Project	-
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										15
(b) Supplementary study in the library, online and in the field										25
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										11
(d) Tutoring										5
(e) Exams and tests										2
(f) Other activities:										-

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	58
3.5 Total hours per semester (3.2+3.4)	100
3.6 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1 Curriculum	-
4.2 Competence	Basic knowledge of functional programming (functional style of writing functions, recursion, types, pattern matching)

5. Requirements (where appropriate)

5.1. For the course	Video projector, blackboard, screen, computer
5.2. For the applications	Computers, equipment and specific software

6. Specific competence

6.1 Professional competences	• analyse big data • analyse business processes • analyse decentralised applications • build predictive models • create data models • debug software • define software architecture • define technical requirements • design cloud architecture • design process for blockchain-based systems • develop blockchain innovative architectures • develop software prototype • develop with cloud services • evaluate blockchain architectures • explain blockchain implications • explain distributed ledger technologies principles • identify blockchain innovation opportunities • implement smart contracts • interpret technical requirements • manage cloud data and storage • outline blockchain-based identity management • oversee development of software • perform data cleansing • perform data mining • perform scientific research • provide technical documentation • recognise blockchain application areas • recognise blockchain risks • use data processing techniques • use software design patterns • use software libraries • utilise computer-aided software engineering tools • utilise machine learning
6.2 Cross competences	The graduate: • develop an analytical approach • taking a proactive approach • developing strategies to solve problems • being open minded • coordinate engineering teams

7. Expected Learning Outcomes

Knowledge	• data analytics • computer programming • ICT debugging tools • software components • software libraries

Skills	The student is able to: • create data sets • design databases in the cloud • develop data processing applications • establish data processes • implement data warehousing techniques • manage data • manage quantitative data • manage research data • perform dimensionality reduction • process data • store digital data and systems • use data processing techniques • use databases • analyse pipeline database information • create data models • analyse decentralised applications • debug software • interpret technical requirements • use software design patterns • use software libraries • adapt to changes in technological development plans
Responsibilities and autonomy	The student has the ability to work independently in order to: • develop an analytical approach • take a proactive approach • develop strategies to solve problems • be open minded

8. Discipline objective (as results from the *key competences gained*)

8.1 General objective	Development of skills and abilities to apply Functional Programming principles in developing and implementing algorithms and software systems in the field of Data Science
8.2 Specific objectives	Gaining knowledge and skills within the context of Data Science aimed to: • understand and use functional concepts • study, design, implement and evaluate applications for Data Science in a functional manner

9. Contents

9.1 Lectures	Hours	Teaching methods	Notes
Abstract Data Types	2	Interactive presentation, students involved in debates.	
Monoids. Functors. Applicatives	2		
Monads and monad transformers	2		
Arrows	2		
Strategies and Eval monad	2		
Arrays, Repa and Par monad for processing large datasets	2		
Accelerate for Graphics Processing Unit (GPU)	2		
Case study: Haskell-based functional languages for blockchain smart contracts: Plutus (Cardano, 3rd generation blockchain platform), Daml (Digital Assets Modeling Language)	2		
Basics of Category Theory	2		
Basics of Category Theory II. Case study: Category Theory for Quantum NLP	2		
Purity of expressions in a quantum program. Case study: Twist Language (MIT new Quantum Computing language)	2		

Compositionality: JAX for high-performance numerical computing	2		
Function type: TensorFlow Federated	2		
Lazy evaluation: Lazy decomposition and conciliation. Robot motion planning	2		

Bibliography:

1. Bryan O'Sullivan, John Goerzen, Don Stewart. Real World Haskell. O'Reilly Media, 2008
2. Simon Marlow. Parallel and Concurrent Programming in Haskell. O'Reilly Media, 2013
3. C. Olah. Neural Networks, Types, and Functional Programming
4. Thinc: A refreshing functional take on deep learning, compatible with your favorite libraries
5. Alexis Toumi. Category Theory for Quantum Natural Language Processing. 2022
6. Xi Cheng, Min Zhou, Xiaoyu Song, Ming Gu, Jiaguang Sun. Parallelizing SMT solving: Lazy decomposition and conciliation. 2018. 10.1016/j.artint.2018.01.001. https://cxcfan.github.io/pub/1-s2.0-S0004370218300237-main.pdf
7. Generalized Lazy Search for Robot Motion Planning: Interleaving Search and Edge Evaluation via Event-based Toggles https://personalrobotics.cs.washington.edu/publications/mandalika2019gls.pdf
8. Charles Yuan, Christopher McNally, Michael Carbin. Twist: Sound Reasoning for Purity and Entanglement in Quantum Programs. Proceedings of the ACM on Programming Languages, Vol 6, Issue POPL, article no. 30m pp 1-32. 2022

9.2 Applications - Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Application design (goals, tools, evaluation metrics)	2	Implementation using specific software tools, experiments and performance evaluation.	
Application Implementation	4		
Testing. Refactoring	2		
Performance Evaluation according to the chosen metrics	3		
Writing the final report	3		

Bibliography:

1. https://docs.discopy.org/en/main/
2. https://www.tensorflow.org/federated/get_started
3. http://www.serpentine.com/criterion/
4. https://github.com/google/jax/blob/main/README.md
5. https://pytorch.org/functorch/stable/

*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

It is carried out through periodic meetings with representatives of the economic environment.

11. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Exam	Written examination	50%
Seminar	-	-	-
Laboratory	-	-	-
Project	Practical abilities to solve the problems and to implement specific applications. Attendance and activity.	Project: continuous assessment (according to the milestones set)	50%
Minimum standard of performance: Both, Written examination and Oral examination, marks are bigger or equal with 5			

Date of filling in: 29.04.2026	Responsible	Title First name Last name	Signature
	Course	Assoc.prof.dr.eng. Radu-Răzvan SLĂVESCU	
	Applications	Assoc.prof.dr.eng. Radu-Răzvan SLĂVESCU	

Date of approval in the department	Head of department, Prof.dr.eng. Rodica Potolea
Date of approval in the Faculty Council	Dean, Prof.dr.eng. Vlad Mureșan