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	Computer science / Master
1.7 Form of education	Full time
1.8 Subject code	2.00

2. Data about the subject

2.1 Subject name				***Algorithms***	
2.2 Course responsible / lecturer				Prof.dr.eng. Potolea Rodica - Rodica.Potolea@cs.utcluj.ro	
2.3 Teacher in charge of seminars / laboratory / project				Assoc.prof.dr.eng. Oprișa Ciprian - Ciprian.Oprisa@cs.utcluj.ro	
2.4 Year of study	I	2.5 Semester	1	2.6 Type of assessment *(E - exam, C - colloquium, V - verification)*	E
2.7 Subject category	Formative category: DA – advanced, DS – speciality, DC – complementary				DS
	Optionality: DI – imposed, DO – optional (alternative), DF – optional (free choice)				DI

3. Estimated total time

3.1 Number of hours per week	3	of which:	Course	1	Seminars	-	Laboratory	1	Project	1
3.2 Number of hours per semester	42	of which:	Course	14	Seminars	-	Laboratory	14	Project	14
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										20
(b) Supplementary study in the library, online and in the field										9
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										20
(d) Tutoring										4
(e) Exams and tests										5
(f) Other activities:										0
3.4 Total hours of individual study (sum (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	Data Structures, Programming Techniques, Algorithms
4.2 Competence	Algorithms performance evaluation, Knowledge of fundamental algorithms on basic data structures

5. Requirements (where appropriate)

5.1. For the course	Blackboard, projector, computer
5.2. For the project	Computers

6. Specific competence

6.1 Professional competences	**C1** - Working with advanced mathematical methods and models, engineering and computing specific techniques and technologies. • **C1.1** – Knowledge and proof of concepts and principles, both theoretical and practical from the software domain. • **C1.2** – Applying specific theory and instruments (algorithms, diagrams, models and tools) for explaining the structure and operation mode of the recent technologies in software, programming environments and programming systems from the scientific literature . • **C1.3** – Using specific models and methods for identifying software components and solutions that are viable for incomplete specifications. **C2** - Development of advanced techniques, methods and methodologies in the domains of software design, programming systems and environments and their applications. • **C2.1** – Identifying and describing the structure and operation mode for complex software systems and applications based on them. • **C2.2** – Building original software components for advanced programming systems, using algorithms, techniques, data structures, technologies and complex programming environments described in the scientific literature. • **C3.3** – Using criteria, techniques and metrics for evaluating and selecting software systems building methodologies, functional and non-functional characteristics.
6.2 Cross competences	**CT3** - Exercising the skill of continuous self-education and demonstrating critical, innovative and research abilities.

7. Discipline objective

7.1 General objective	Acquiring problem-solving skills specific to data analysis tasks, identifying potential solutions and selecting appropriate ones.
7.2 Specific objectives	• Complexity computation • Distinguishing between complexity classes • Approximating computationally hard problems • Knowledge of algorithms specific to big data

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Algorithms complexity	2	Discussion and multimedia materials Interractive teaching style Involving students in presenting some case studies Problems solving	N/A
Complexity classes; P vs. NP, NP-Completeness	2		
NP-complete problems; reduction algorithms	2		
NP-complete problems approximation	2		
Sublinear algorithms for big data	2		
Linear programming	2		
Searching in big data	2		

Bibliography

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
2. Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.
3. Feldman, Moran. *Algorithms for Big Data*. World Scientific, 2020.

8**.2 Applications – seminars / laboratory / project**	Hours	Teaching methods	Notes
Experimental analysis of the algorithms complexity	2	Discussion and multimedia materials. Study of relvant research papers.	N/A
Identifying the algorithms complexity classes	2		
Reduction algorithms for NP-Complete problems	2		
Approximation algorithms for NP-Complete problems	2		

Implementation of sublinear algorithms	2	Applying the studied techniques.	
Implementation and analysis of linear programming techniques	2		
Algorithms for searching in big data	2		
Introduction and presentation of the project theme	2		
Bibliographic research: resources identification	2		
Bibliographic research: documentation analysis and synthesis	2		
Requirements gathering	2		
Implementing and testing a case study	2		
Analysis of the experimental results	2		
Redacting and presenting the final document	2		

Bibliography:

1. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
2. Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.
3. Feldman, Moran. *Algorithms for Big Data*. World Scientific, 2020.

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

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Theoretical analysis and problem solving skills based on the acquired knowledge	Written exam	50%
Laboratory	Based on the obtained results and the analysis and synthesis skills	Practical problem-solving skills	20%
Project	Based on the obtained results and the analysis and synthesis skills	Oral evaluation	30%

Minimum standard of performance:

- Laboratory ≥ 5
- Project evaluation ≥ 5
- Final exam ≥ 5;

Final grade computation: 50% final exam + 20% laboratory + 30% project

Date of filling in: 26.02.2025	**Responsible**	**Title, First name Last name**	**Signature**
	Course	Prof.dr.eng. Rodica POTOLEA	
	Applications	Assoc.prof.dr.eng. Ciprian OPRIȘA	

Date of approval in the department 17.09.2025	Head of department, Prof.dr.eng. Rodica Potolea
Date of approval in the Faculty Council 17.09.2025	Dean, Prof.dr.eng. Vlad Mureșan