SYLLABUS

Algorithms, models and concepts in distributed systems

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babes-Bolyai University
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Master
1.6. Study programme/Qualification	Databases
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline	**Algorithms, models and concepts in distributed systems**					Discipline code	**MME8110**
2.2. Course coordinator				Assoc. prof. Rareş Florin Boian, Ph.D.			
2.3. Seminar coordinator				Assoc. prof. Rareş Florin Boian, Ph.D.			
2.4. Year of study	1	2.5. Semester	2	2.6. Type of evaluation	E	2.7. Discipline regime	Optional

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**4**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**2**
3.4. Total hours in the curriculum	48	of which: 3.5 course	24	3.6 seminar/laboratory/project	**24**
Time allotment for individual study (ID) and self-study activities (SA)					**hours**
Learning using manual, course support, bibliography, course notes (SA)					20
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					20
Tutorship					10
Evaluations					9
Other activities:					
3.7. Total individual study hours	**79**				
3.8. Total hours per semester	**175**				
3.9. Number of ECTS credits	**7**				

4. Prerequisites (if necessary)

4.1. curriculum	
4.2. competencies	

5. Conditions (if necessary)

5.1. for the course	
5.2. for the seminar /lab activities	

6.1. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	• use of theoretical foundations of computer science as well as of formal models • design and administration of computer networks
Transversal competencies	• application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms • use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language

6.2. Learning outcomes

Knowledge	• The graduate has the necessary knowledge for using computers, developing software programs and applications, information processing. • The graduate has the necessary knowledge for literature review.
Skills	• The graduate has the ability to develop, design and create new applications, systems or products using best practices of the field. • The graduate has the necessary skills for computer program design and software systems analysis. • The graduate has the necessary skills to use research support tools.
Responsibility and autonomy:	• The graduate is able to identify complex problems and examine related issues to develop solving options and implement solutions. • The graduate is able to combine diverse information to formulate solutions and generate ideas for developing new products and applications. • The graduate is able to apply architectural styles, design patterns and best practices in the field to design software applications of high complexity. • The graduate has the ability to evaluate different architectures and possible solutions to a problem and choose the right one for the specific requirements and constraints of the application to be developed.

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Learning basic distributed systems and distributed algorithms concepts • Learn to implement distributed algorithms
7.2 Specific objective of the discipline	• Abstractions used in modelling the distributed algorithms • Distributed systems theoretical models • Broadcast algorithms • Shared memory algorithms • Consensus algorithms

8. Content

8.1 Course	Teaching methods	Remarks

Weeks 1-2: Distributed systems models and abstractions	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Weeks 3-4: Basic and reliable broadcast algorithms	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Weeks 5-6: Uniform and probabilistic broadcast algorithms	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Weeks 7-8: Shared memory – regular registers	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Weeks 9-10: Shared memory – atomic registers	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Weeks 11-12: Consensus - flooding	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Bibliography 1. CHRISTIAN CACHIN, RACHID GUERRAOUI, LUIS RODRIGUES, Introduction to Reliable and Secure Distributed Programming, Second Edition, Springer, 2011 2. LYNCH N.A. Distributed Algorithms. Morgan Kaufmann Pub. 1996 3. Ajay D. Kshemkalyani, Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2011 4. BOIAN F.M. Programarea distribuita in internet; metode si aplicatii. Ed. Albastra, Cluj, 1997 5. TANENBAUM A.S. Distributed Operating Systems. Prentice Hall, 2000 6. TEL G. Introduction to Distributed Algorithms. Cambridge Press, 1994		
8.2 Seminar / laboratory	Teaching methods	Remarks
Distributed algorithm implementation architecture	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Detailed discussion about the implementation and testing of the broadcast algorithm	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Detailed discussion about the implementation and testing of the shared memory algorithm	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Detailed discussion about the implementation and testing of the consensus algorithm	• Interactive exposure • Explanation • Conversation • Didactical demonstration	
Bibliography 1. CHRISTIAN CACHIN, RACHID GUERRAOUI, LUIS RODRIGUES, Introduction to Reliable and Secure Distributed Programming, Second Edition, Springer, 2011		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- By learning the theoretical and methodological concepts and addressing the practical aspects of the Algorithms, models and concepts in distributed systems course, students acquire a body of knowledge consistent, consistent with partial competencies required for possible occupations provided in Grid 1 - RNCIS
- The course complies with IEEE and ACM Curriculla Recommendations for Computer Science studies.
- The course curriculum exists in universities and faculties in Romania
- The course content is very well appreciated by software companies whose employees and graduates of this course

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	The level of knowledge and understanding of the course subjects	Written exam	30%
10.5 Seminar/laboratory	Ability to solve practical problems, specific to the course subjects	Shared memory project	30%
		Consensus project	30%
10.6 Minimum standard of performance			
• Minimum 5 in the final grade			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:
...

Signature of course coordinator

.....

Signature of seminar coordinator

.....

Date of approval:
...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable*.".