SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	Babeş-Bolyai University
1.2 Faculty	Faculty of Mathematics and Computer Science
1.3 Department	Department of Computer Science
1.4 Field of study	Computer Science
1.5 Study cycle	Master
1.6 Study programme / Qualification	High Performance Computing and Big Data Analytics

2. Information regarding the discipline

2.1 Name of the discipline	Formal Models of Concurrency						
2.2 Course coordinator	Lect. PhD. Sterca Adrian						
2.3 Seminar coordinator	Lect. PhD. Sterca Adrian						
2.4. Year of study	**1**	2.5 Semester	**1**	2.6. Type of evaluation	**E**	2.7 Type of discipline	**Compulsory**

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

3.1 Hours per week	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					30
Additional documentation (in libraries, on electronic platforms, field documentation)					40
Preparation for seminars/labs, homework, papers, portfolios and essays					40
Tutorship					28
Evaluations					20
Other activities:					0

3.7 Total individual study hours	158
3.8 Total hours per semester	175
3.9 Number of ECTS credits	7

4. Prerequisites (if necessary)

4.1. curriculum	• Distributed operating systems, Concurrent and distributed programming
4.2. competencies	• Concurrent and distributed programming

5. Conditions (if necessary)

5.1. for the course	• Class room with a video projector device
5.2. for the seminar /lab activities	•

6. Specific competencies acquired

Professional competencies	• formal models for concurrent processes • concurrent programming paradigms
Transversal competencies	▪ ability to model and analyze concurrent processes ▪ ability to evaluate the performance of a concurrent system

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

7.1 General objective of the discipline	• The course offers a strong theoretical approach in modelling concurrent and distributed systems
7.2 Specific objective of the discipline	• The course presents several formalisms for describing concurrent processes

8. Content

8.1 Course	Teaching methods	Remarks
1. Serialization, concurrency and parallelism	Exposure:description, explanation,examples, discussion of case studies	
2. Concurrent programming paradigms: shared memory, message passing, transactional memory	Exposure:description, explanation,examples, discussion of case studies	
3. Concurrency control – serializability criteria	Exposure:description, explanation,examples, discussion of case studies	
4. Concurrency control algorithms	Exposure:description, explanation,examples, discussion of case studies	
5. Process algebra I	Exposure:description, explanation,examples, discussion of case studies	
6. Process algebra II	Exposure:description, explanation,examples, discussion of case studies	
7. CCS - Calculus of Communicating Systems	Exposure:description, explanation,examples,	

	discussion of case studies	
8. CCS (part 2)	Exposure:description, explanation,examples, discussion of case studies	
9. CCS (part 3)	Exposure:description, explanation,examples, discussion of case studies	
10. CCS (part 4)	Exposure:description, explanation,examples, discussion of case studies	
11. Pi-calculus	Exposure:description, explanation,examples, discussion of case studies	
12. Other formalisms: CSP – Communicating Sequential Processes, the Actor model, PEPA	Exposure:description, explanation,examples, discussion of case studies	
13. Other formalisms: CSP – Communicating Sequential Processes, the Actor model, PEPA	Exposure:description, explanation,examples, discussion of case studies	
14. Current research trends in formal modelling of concurrency	Exposure:description, explanation,examples, discussion of case studies	

Bibliography

1. Robin Milner, *Communication and Concurrency*, Prentice Hall, International Series in Computer Science, ISBN 0-13-115007-3. 1989
2. Reichel H. Formal Models of Concurrency,http://www.informatik.uni-bremen.de/~lschrode/teaching/Systems/ReichelCCS.pdf, 2003.
3. C. A. R. Hoare, Communicating Sequential Processing, 2004, http://www.usingcsp.com/cspbook.pdf
4. Weikum G., Vossen G., Transactional Information System: Theory, Algorithms, and Practice of Concurrency Control and Recovery. Kaufmann Morgan Publ. 2002.

8.2 Seminar / laboratory	Teaching methods	Remarks
1. Discussions related to the project or report's subject	Dialog, debate, case studies, examples	
2. Discussions related to the project or report's subject	Dialog, debate, case studies, examples	
3. Discussions related to the project or report's subject	Dialog, debate, case studies, examples	
4. Report presentations	Dialog, debate, case studies, examples	
5. Report presentations	Dialog, debate, case studies, examples	
6. Project presentations	Dialog, debate, case studies, examples	
7. Project presentations	Dialog, debate, case studies, examples	

Bibliography: recent articles from ACM Digital Library and IEEE Xplore

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

- The course respects the IEEE and ACM Curriculla Recommendations for Computer Science studies;

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Knowing the formalisms for describing concurrency presented during the course	Examination	30 %
10.5 Seminar/lab activities	Ability to understand recent research and relate it to theoretical concepts presented at the course	Project Research report	40 % 30 %
10.6 Minimum performance standards			
In order to successfully pass this course, students must get at least 5 at each of the 3 examination tasks.			

Date	Signature of course coordinator	Signature of seminar coordinator
.....	Lect.PhD. Adrian Sterca	Lect.PhD. Adrian Sterca

Date of approval — Signature of the head of department

..... — Prof. PhD. Bazil Parv