



DISCIPLINE ACADEMIC SHEET
ACADEMIC YEAR 2025 – 2026

1. PROGRAMME DATA

1.1. Higher Education Institution	UNIVERSITY OF CRAIOVA
1.2. Faculty	Automation, Computers and Electronics
1.3. Department	Computers and Information Technology
1.4. Field of Study	Computers and Information Technology
1.5. Study level	L(License / Undergraduate)
1.6. Organizational form	IF
1.7. Study Program (name/code) /Qualification	Computers / L2060101010

2. DISCIPLINE DATA

2.1 Discipline Name	Concurrent and Distributed Systems						
2.2 Course Activities Holder	Costin BĂDICĂ						
2.3. Practical Activities Holder							
2.4. Study Year	3	2.5. Semester	1	2.6. Evaluation type	E	2.7. Discipline regime	DOB

3. ESTIMATED TOTAL TIME (hours per semester of teaching activities)

3.1. Number of hours per week	5	in which: 3.2 course	3	3.3. sem./lab./pr.	2
3.4. Total hours of curriculum	70	in which: 3.5 course	42	3.6. sem./lab./pr.	28
3.7. Time distribution - hours/week					hours
Study after manual, course support, bibliography and notes					20
Additional documentation in library, on specialized electronic platforms and field					10
Training seminars / labs, homework, portfolios and essays					15
Tutorial					5
Examinations					5
Other activities.....					-
Total hours per individual activities					55
3.8. Total hours per semester					125
3.9. Number of credits					5

4. PRECONDITIONS (where appropriate)

4.1. of curriculum	• Object-Oriented Programming. Computer Architecture.
4.2. of competence	• n/a

5. CONDITIONS (where appropriate)

5.1. of the course	• The course is taught using interactive face-to-face (f2f) and/or online lectures, distribution of course materials in digital format and interaction with the students. The
--------------------	---

	teaching process is structured as follows: - 70% theoretical presentations, using the course support (slides), as well as examples of theoretical exercises and practical demonstrations and tutorials. - 30% interaction (discussions with the students) based on questions,
5.2. of seminar/ laboratory/project	•

6. COURSE OBJECTIVES – expected learning outcomes that are supported by completing and passing the course

Knowledge	The student/graduate: 1. Identifies and describes fundamental concepts of concurrent and distributed systems, principles of process synchronization and coordination. 2. Explains and interprets theoretical and experimental results from computer science regarding concurrent and distributed programming. 3. Describes, identifies, summarizes elementary concepts and methods regarding concurrent and distributed systems and the way they are applied to concrete problems. 4. Describes, identifies, summarizes elementary concepts and methods regarding programming languages, programming environments, concurrent and distributed programming techniques and the way they are applied to concrete problems. 5. Describes, identifies, summarizes elementary concepts and methods regarding the modeling, analysis, design and testing of concurrent and distributed computing systems. 6. Describes and identifies models for different components of concurrent and distributed computing systems and evaluates the functional and non-functional characteristics. 7. Describes and identifies specific methods and tools for the analysis, design and implementation of concurrent and distributed systems.
Skills (Abilities)	The student/graduate operates with specific methods for the design, development and analysis of distributed applications, using dedicated software tools. The student/graduate solves performance, security and reliability problems in concurrent and distributed systems, validating the proposed solutions.
Responsibility and autonomy	The student/graduate behaves honorably, responsibly and ethically, in the spirit of the law, to ensure the reputation of the profession.

7. CONTENT

7.1. Course	The method of implementation	Teaching methods	Allocated time (hours)
1. The model of concurrent computation		Lectures. Theoretical examples. Practical demonstrations.	3
2. Critical sections			3
3. Verification of concurrent programs		Interaction through questions and answers.	3
4. Semaphores			3
5. Monitors			3
6. Concurrent objects			3
7. Linearizability			3
8. Concurrent data structures			3
9. Solving concurrent problems			3
10. Channels and message passing			3
11. Introduction to distributed			3

systems			
12. Inter-process communication in distributed systems			3
13. Object-oriented distributed systems and remote method invocation			3
14. Exam review			3
Bibliography:			
1. M. Ben-Ari, Principles of Concurrent and Distributed Programming, 2/E, Addison-Wesley, 2006 2. Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming, 2/E, Elsevier, 2020 3. Jeff Magee, Jeff Kramer, Concurrency: State Models & Java Programs, 2nd Edition, John Wiley & Sons, 2006 4. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Distributed Systems. Concepts and Design (5th edition), Addison-Wesley, 2011 5. Scott Oaks, Henry Wong, Java Threads (3rd ed.). Understanding and Mastering Concurrent Programming. O'Reilly Media. 2009			

7.2. Seminar/laboratory/project	The method of implementation	Teaching methods	Allocated time (hours)
1. Introduction. The laboratory requirements		Tutorials. Practical demonstrations. Interaction through questions and answers	2
2&3: The basics of thread programming. The Java threads API.			4
4. Critical sections and synchronization.			2
5. Implementing critical sections in Java			2
6. Implementing semaphores			2
7. Monitors and condition-based synchronization			2
8. Explicit locks			2
9. Thread pools			2
10. The Java executors API			2
11. Inter-process communication. The Java API for UDP and TCP			2
12. Group communication			2
13. Remote invocation			2
14. Evaluation and make-up laboratories			2
Bibliography:			
1. M. Ben-Ari, Principles of Concurrent and Distributed Programming, 2/E, Addison-Wesley, 2006			
2. Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming, 2/E, Elsevier, 2020			
3. Jeff Magee, Jeff Kramer, Concurrency: State Models & Java Programs, 2nd Edition, John Wiley & Sons, 2006			
4. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Distributed Systems. Concepts and Design (5th edition), Addison-Wesley, 2011			
5. Scott Oaks, Henry Wong, Java Threads (3rd ed.). Understanding and Mastering Concurrent Programming. O'Reilly Media. 2009			

8. Correlating the content of the discipline with the expectations of representatives from the epistemic community, professional associations and representative employers in the field related to the program

The content of the discipline is meticulously correlated with the expectations of the epistemic community (researchers) regarding concurrency, which constitutes the largest part of the curriculum. The main emphasis is placed on achieving an in-depth mastery of threading and multiprocessing, together with fundamental process synchronization techniques (including locks, semaphores and monitors) and the management of shared memory and state in order to develop highly efficient and thread-safe applications. The last part covers essential distributed systems concepts, such as Inter-Process Communication (IPC) and fault tolerance principles. This curriculum guarantees that graduates acquire the critical and highly sought-after skills in modern concurrency models, allowing them to design and implement scalable, reliable and high-performance systems, thus directly addressing the industry's needs for expertise in core systems engineering.

9. EVALUATION

Activity Type	9.1. Evaluation criteria	9.2. Evaluation methods	9.3. Final mark weight
9.4. Course	Solving exercises and problems involving methods and techniques specific to concurrent and distributed systems.	Written exam	70 %
9.5. Seminar/laboratory/project	Practical results during the laboratory works	The acquired practical knowledge is continuously evaluated during the laboratory sessions, by assessing the practical results and the answers to questions.	30 %
9.6. Minimum standard of performance			
Minimum grade 5.			

Date of Completion	Signature of the Course Activities Holder,	Signature of the Practical Activities Holder,
23.08.2025	Costin BĂDICĂ	

Date of Approval in the Department	Signature of the Department Director,
25.08.2025	Ș.I.dr.ing. Nicolae ENESCU

Date of Approval in the Faculty	Signature of the Dean,
	Prof.dr.ing. Dorin ŞENDRESCU