



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	Parallel and Distributed Algorithms						
2.2 Course Activities Holder	Mihai L. MOCANU, Ph.D., Professor						
2.3. Practical Activities Holder	Constantin Renato IVĂNESCU, Ph.D., Lecturer						
2.4. Study Year	III	2.5. Semester	II	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					30/14
Study after manual, course support, bibliography and notes					7
Additional documentation in library, on specialized electronic platforms and field					9
Training seminars / labs, homework, portfolios and essays					7
Tutorial					1
Examinations					6
Other activities.....					-
Total hours per individual activities					30
3.8. Total hours per semester					100
3.9. Number of credits					4

4. PRECONDITIONS (where appropriate)

4.1. of curriculum	Basic knowledge acquired during the previous years of study are necessary, in: – Procedural and structured programming – Programming methods and techniques – Algorithms and Data Structures – Object Oriented Programming – Computer Architecture – Operating Systems
4.2. of competence	None

5. CONDITIONS (where appropriate)

5.1. of the course	Lectures are held in university lecture halls equipped with the necessary audio-visual equipment. The primary curriculum relies on a textbook authored by the course coordinator, originally published in 1995 and updated on an annual basis. Students can access all essential materials—including documentation, slide decks, past exams, quizzes, and seminar problems—via the online course portal. The instructional design emphasizes active learning, featuring interactive concept discussions at the conclusion of each class, monthly evaluation quizzes, and comprehensive chapter homework designed to reinforce fundamental concepts.
5.2. of seminar/ laboratory/project	Laboratory activities are conducted in a dedicated facility, operating under a strict one-student-per-workstation ratio to ensure individual, practical engagement. Under the guidance of a qualified instructor, students learn to program and utilize the specific functionalities of parallel computing architectures. The software environments used during practical hours are based on PVM, MPI, and OpenMP. Coursework is aligned with an official laboratory manual, originally published in 1998 and updated annually to maintain technical relevance; this material is fully accessible to all students. Formative evaluation includes mandatory homework assignments tailored to each topical module, enforcing a standardized two-week submission window for student solutions.

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

Knowledge	Student/Graduate: 1. Describes, identifies, and summarizes elementary concepts and methods regarding programming languages, programming environments, programming techniques databases. artificial intelligence and software engineering and their application in concrete problems.
Skills (Abilities)	Student/Graduate: 1. Chooses and explains own concepts specific to algorithm design, object-oriented programming, logical and functional programming. The student/ graduate specifies requirements, analyzes, designs, develops and tests software systems using general-purpose and specialized (parallel) programming language environments, applying the specific elements of software engineering..
Responsibility and autonomy	Student/Graduate: 1. Applies the values of ethics and deontology of the engineering profession, practices logical reasoning, evaluation and self-evaluation in decision-making. 2. Has an honorable, responsible, ethical behavior, 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: Introduction. Parallel Architectures. Virtual Platforms.	Synchronous online	Lectures will be delivered through interactive presentations using computers and specialized software, such as PowerPoint and Integrated Programming Environments (IDEs). Quizzes and discussions are scheduled to take place at the end of each lecture.	3
2: Distributed Architectures. Parallel and Distributed Architectural Models.	Synchronous online		3
3: Parallel Programming Models. Performance Metrics.	Synchronous online		3
4: Parallel Algorithms: Concurrency, Mapping and Basic Communication.	Synchronous online		3

5: Parallel Algorithms: Design Principles	Synchronous online	Practical experience with the C programming language will be emphasized during all activities, and updates and improvements will be noted..	3
6: Parallel Numerical Alg. (1). Pipeline/ Systolic Methods in Matrix Computations	Synchronous online		3
7: Parallel Numerical Algorithms (2). Linear Equation Systems	On site (face to face)		3
8: Parallel Sorting	On site (face to face)		3
9: Parallel Algorithms for Graphs	On site (face to face)		3
10: Parallel Algorithms for Binary Trees	On site (face to face)		3
11: Synchronous Distributed Algorithms: Atomicity and Consensus.	On site (face to face)		3
12: Asynchronous Dist.Alg.: Termination Detection, Asynchronous Replication	On site (face to face)		3
13: Time in Distributed Systems: Logical Time, Vector Time.	On site (face to face)		3
14: Routing Algorithms	On site (face to face)		3

Bibliography (A-Core text, B-Secondary text, C-peripheral reading):

[A1] V. Kumar, A. Grama, A. Gupta, G. Kyrpys - Introduction to Parallel Computing Benjamin/Cummings 2003 (*a globally recognized foundational text for structural mappings of parallel architectures, message passing, and network routing topologies*)

[A2] P. Pacheco and M. Malensek, An Introduction to Parallel Programming, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2021. (*the modern standard curriculum text for hands-on, portable parallel programming using updated specifications of MPI, OpenMP, and Pthreads*)

[B1] R. W. Hockney, C.R. Jesshope. Parallel Computers. Architecture, Programming, Algorithms, Ed. Tehnica, 1991

[B2] D. Grigoras – Parallel Computing. From Systems to Applications, Computer Libris Agora, 2000

[A3] M. Mocanu, Algoritmi și limbaje de programare paralele (manual de curs, Reprografia Univ. Craiova, 1995), online la (<http://software.ucv.ro/~mmocanu/ro/resurse/Calcul%20paralel/Docs/indexL.html>)

[A4] Akl S., The Design and Analysis of Parallel Algorithms (Prentice-Hall, 1989)

[C1] Chaudhuri P., Parallel Algorithms Design and Analysis (Prentice-Hall, 1992)

[C2] JaJa J., An Introduction to Parallel Algorithms (Addison Wesley, 1992)

[C3] Galea D., Brudaru O., An Introduction to Systolic Computation (Ed. Academiei, Bucuresti, 1994)

[B3] Hoare C.A.R., Communicating Sequential Processes, Prentice-Hall 1985

[B4] S. Mullender (ed.), Distributed Systems. Addison-Wesley, 1993

[A5] N. Lynch. Distributed Algorithms. Morgan Kaufmann, 1996 (*definitive mathematical authority on designing, building, and formally verifying fault-tolerant decentralized computing models*)

[A6] G. Tel., Introduction to Distributed Algorithms. Cambridge Univ. Press, 2000 (an instructional guide focused heavily on fundamental message-passing protocols and distributed control structures)

7.2. Seminar/laboratory/project	The method of implementation	Teaching methods	Allocated time (hours)
Laboratory:			
1: Introduction. Parallel Models. Execution Parallelism	Synchronous online	Problem implementation, presentation according to the topic of each weekly session. Integration of all knowledge in a semester homework (hardware/ software/ integrative “project”). The topic of the project is taken from a list proposed by the instructor at the debut of the semester. Project de-livery is accompanied by report and presentation.	2
2: Parallel Programming using PVM (Parallel Virtual Machine). The Structure of the PVM Environment. PVM Process Control.	Synchronous online		2
3: The Implementation of Parallel Applications using PVM	Synchronous online		2
4: Parallel Programming using MPI (Message Passing Interface). The Structure of the MPI Environment. Interfaces. The Static Model. Topologies.	Synchronous online		2
5: The Implementation of Parallel Applications using MPI.	Synchronous online		2
6: Unix Processes. Concurrency	Synchronous online		2

Support. Parallel Process Synchronization.			
7: The Implementation of a Support for Process Parallel Execution.	On site (face to face)		2
8: Parallel Programming using Threads in Unix.	On site (face to face)		2
9: Thread Synchronization. The Producer-Consumer Problem.	On site (face to face)		2
10: Parallel Programming using Threads in Windows.	On site (face to face)		2
11: Windows Threads Applications	On site (face to face)		2
12: Thread Implementation. Errors and Exceptions.	On site (face to face)		2
13: Software Metrics. Testing using Specialized Programs.	On site (face to face)		2
14: Performance Testing. Benchmarks.	On site (face to face)		2
Bibliography: Some of the references above are also used, plus internal documentations for programmers. [B5] C. H. Nevison et al., Laboratories for Parallel Computing. Boston, MA, USA: Jones and Bartlett Publ., 1994. <i>(a specialized historical lab manual that introduces hands-on parallel programming environments and foundational exercises using early message-passing paradigms).</i>			

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

Course and application contents are in accordance with similar items from other universities in the country and abroad. Gained competences are convergent with the requirements of major employers, being the result of periodic discussions with representatives of the local companies active in the IT field.

9. EVALUATION

Activity Type	9.1. Evaluation criteria	9.2. Evaluation methods	9.3. Final mark weight
9.4. Course	Concepts understanding; problem correct solving; ability to connect concepts as they are taught; gain of analysis and synthesis skills.	Written/ oral verification Quizzes Homework	40% 20% 20%
9.5. Seminar/laboratory/project	Individual activity but also as a team member.	$\text{NotaLab} = 50\% * (\text{Nota1} + \text{Nota2}) + 50\% * \text{Nota3}$ Nota1: Mark of teamwork Nota2: Mark for individual presentation of own work in semester team project Nota3: Mark for individual activity during classes	20%
9.6. Minimum standard of performance Basic knowledge in solving problems using a parallel algorithmic language. Active participation during at least 12 (of 14) laboratory sessions and proof of implementation abilities. The student must obtain at least 50% (min. 10 pts.) during intermediary evaluations (quizzes, homework, laboratory assignments) and also at least 50% (min. 20 pts) at the final written exam, to promote			

Date of Completion	Signature of the Course Activities Holder,	Signature of the Practical Activities Holder,
1.10.2025		

Date of Approval in the Department	Signature of the Department Director,
6.10.2025	Ș.l.dr.ing. Nicolae ENESCU

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