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	Bachelor of Science
1.6 Program of study/Qualification	Computer science/ Engineer
1.7 Form of education	Full time
1.8 Subject code	54.2

2. Data about the subject

2.1 Subject name	***Parallel Programming***				
2.2 Course responsible/lecturer	Prof. dr. eng. Alin Suciu – alin.suciu@cs.utcluj.ro				
2.3 Teachers in charge of seminars/ laboratory/ project	Prof. dr. eng. Alin Suciu – alin.suciu@cs.utcluj.ro				
2.4 Year of study	IV	2.5 Semester	2	2.6 Type of assessment (E - exam, C - colloquium, V - verification)	E
2.7 Subject category	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

3. Estimated total time

3.1 Number of hours per week	5	of which:	Course	2	Seminars	1	Laboratory	2	Project	
3.2 Number of hours per semester	70	of which:	Course	28	Seminars	14	Laboratory	28	Project	
3.3 Individual study:										
(a) Manual, lecture material and notes, bibliography										28
(b) Supplementary study in the library, online and in the field										22
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										26
(d) Tutoring										0
(e) Exams and tests										4
(f) Other activities:										0

3.4 Total hours of individual study (suma (3.3(a)…3.3(f)))	80
3.5 Total hours per semester (3.2+3.4)	150
3.6 Number of credit points	6

4. Pre-requisites (where appropriate)

4.1 Curriculum	Computer Programming (C), OO Programming (Java/C#), Logic Programming (Prolog), Operating Systems
4.2 Competence	All competences related to the above disciplines

5. Requirements (where appropriate)

5.1. For the course	Blackboard, Projector, Computer, Online platforms
5.2. For the applications	Multicore computers, Specific Software, Online platforms

6. Specific competence

6.1 Professional competences	**C3** - Problems solving using specific Computer Science and Computer Engineering tools (1 credit) **C3.1** Identifying classes of problems and solving methods that are specific to computing systems **C3.2** Using interdisciplinary knowledge, solution patterns and tools, making experiments and interpreting their results **C3.3** Applying solution patterns using specific engineering tools and mehods

	C3.4 Comparatively and experimentaly evaluation of the alternative solutions for performance optimization **C3.5** Developing and implementing informatic solutions for concrete problems **C5 -**Designing, managing the lifetime cycle, integrating and ensuring the integrity of hardware, software and communication systems (1 credit) **C5.1** Specifying the relevant criteria regarding the lifetime cycle, quality, security and computing system's interaction with the environment and human operator **C5.2** Using interdisciplinary knowledge for adapting the computing system to the specific requirements of the application field **C5.3** Using fundamental principles and methods for security, reliability and usability assurance of computing systems **C5.4** Adequate utilization of quality, safety and security standards in information processing **C5.5** Creating a project including the problem's identification and analysis, its design and development, also proving an understanding of the basic quality requirements **C6 -** Designing intelligent systems (2 credits) **C6.1** Describing the components of intelligent systems **C6.2** Using domain-specific tools for explaining and understanding the functioning of intelligent systems **C6.3** Applying the fundamental methods and principles for specifying solutions for typical problems using intelligent **C6.4** Choosing the criteria and evaluation methods for the quality, performances and limitations of intelligent systems **C6.5** Developing and implementing professional projects for intelligent systems
6.2 Cross competences	N/A

7. Discipline objective (as results from the *key competences gained*)

7.1 General objective	Developing the ability to identify parallelism in a given problem, and to take advantage of this parallelism using various methods and technologies for parallel programming
7.2 Specific objectives	▪ Understanding the performance parameters of parallel algorithms ▪ Ability to implement parallel algorithms using multithreading technologies (in C, Java, C#, Prolog, OpenMP) ▪ Ability to implement parallel algorithms based on the VSM model (Linda) ▪ Ability to implement parallel algorithms based on message passing (PVM, MPI) ▪ Basic knowledge of the cutting edge developments in the field (Quantum Computing, DNA Computing)

8. Contents

8.1 Lectures	Hours	Teaching methods	Notes
Introduction, Types of Parallelism, Classification, Applications	2	Lectures using blackboard and projector, interactive discussions. Online platforms	
Parallel Algorithms, Performance Parameters, Amdahl's Law, Gustafson's Law	2		
Processes (C/UNIX), Communication, Synchronization	2		
Threads (Java, C#, Prolog), Communication, Synchronization	2		
OpenMP (1)	2		
OpenMP (2)	2		
OpenMP (3)	2		
Linda, Parallelism based on Virtual Shared Memory	2		
Message Passing Programming, PVM, MPI	2		
Programming the Graphics Processor (GPU)	2		
Sorting Networks	2		

Cryptography and Cryptanalysis concepts	2		
Grid Computing, Cluster Computing	2		
Quantum Computing and DNA Computing	2		

Bibliography

1. Peter Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann, 2011.
2. Barbara Chapman, Gabriele Jost and Ruud van der Pas, Using OpenMP - Portable Shared Memory Parallel Programming, MIT Press, 2007 (online).
3. I. Foster, Designing and Building Parallel Programs, Addison Wesley, 1995 (online).
4. L. Sterling, E. Shapiro, The Art of Prolog, MIT Press, 1994.

8.2 Applications – Seminars/Laboratory/Project	Hours	Teaching methods	Notes
Imperative Programming in C – review, Solving highly parallelizable problems	2	Practical laboratory works / programming exercises using specific software tools and Online platforms	
Logic Programming in Prolog – review, Solving highly parallelizable problems	2		
Processes (C/UNIX)	2		
Threads (C)	2		
Threads (Java, C#)	2		
Threads (Prolog)	2		
Programming in OpenMP (1)	2		
Programming in OpenMP (2)	2		
Programming in OpenMP (3)	2		
Programming in Linda	2		
Programming in MPI	2		
Sorting Networks	2		
Cryptographic Algorithms	2		
Final Evaluation	2		

Bibliography

1. Peter Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann, 2011.
2. Barbara Chapman, Gabriele Jost and Ruud van der Pas, Using OpenMP - Portable Shared Memory Parallel Programming, MIT Press, 2007 (online).
3. I. Foster, Designing and Building Parallel Programs, Addison Wesley, 1995 (online).
4. L. Sterling, E. Shapiro, The Art of Prolog, MIT Press, 1994.

** Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

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

The content of the course is aligned to the latest developments in the field and responds to both the development in the multicore / other parallel hardware technologies and the requirements coming from the industry.

10. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Knowledge assimilated from the course material, interactivity during lectures. Ability to solve domain specific problems	Written and/or online exam using online platforms (E)	50%
Seminar	Ability to solve domain specific problems	Written test and/or Seminar homeworks sent/ received via online platforms (S)	20%
Laboratory	Ability to solve problem using parallel programming techniques and technologies	Written test and/or Laboratory homeworks sent/ received via online platforms (L)	30%
Project			

Minimal performance requirements: E ≥ 50% ; L ≥ 50%
Final Grade: N = 0.5*E + 0.2*S + 0.3*L

Date of filling in:	Titulari	Titlu Prenume NUME	Semnătura
	Course	Prof.dr.eng. Alin Suciu	
	Applications	Prof.dr.eng. Alin Suciu	

Date of approval in the department	Head of department Prof.dr.ing. Rodica Potolea
Date of approval in the Faculty Council	Dean Prof.dr.ing. Liviu Miclea