COURSE DESCRIPTION

1. Program identification information

1.1 Higher education institution	**National University of Science and Technology Politehnica Bucharest**
1.2 Faculty	**Electronics, Telecommunications and Information Technology**
1.3 Department	**Electronic Devices, Circuits and Architectures**
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technology
1.5 Cycle of studies	Masters
1.6 Programme of studies	Advanced Computing in Embedded Systems

2. Date despre disciplină

2.1 Course name (ro) (en)	Calcul paralel Parallel Computing
2.2 Course Lecturer	S.l./Lect. Dr. George Valentin STOICA
2.3 Instructor for practical activities	S.l./Lect. Dr. George Valentin STOICA

2.4 Year of studies	2	2.5 Semester	1	2.6. Evaluation type	E	2.7 Course regime	Ob

2.8 Course type	DA	2.9 Course code	UPB.04.M3.O.26-24	2.10 Tipul de notare	Nota

3. Total estimated time (hours per semester for academic activities)

3.1 Number of hours per week	4	Out of which: 3.2 course	2	3.3 seminary/laboratory	2
3.4 Total hours in the curricula	56	Out of which: 3.5 course	28	3.6 seminary/laboratory	28

Distribution of time:	hours
Study according to the manual, course support, bibliography and hand notes Supplemental documentation (library, electronic access resources, in the field, etc) Preparation for practical activities, homework, essays, portfolios, etc.	65
Tutoring	0
Examinations	4
Other activities (if any):	0

3.7 Total hours of individual study	69.00
3.8 Total hours per semester	125
3.9 Number of ECTS credit points	5

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	Not applicable
4.2 Results of learning	Not applicable

5. Necessary conditions for the optimal development of teaching activities (where applicable)

5.1 Course	The course will take place in a room equipped with a video projector.

5.2 Seminary/ Laboratory/Project	The laboratory will take place in a class with specific equipment, which must include: PCs, software development environment such as MS Visual Studio, multicore CPU and/or GPU such as nVidia GTX

6. General objective *(Reffering to the teachers' intentions for students and to what the students will be thought during the course. It offers an idea on the position of course in the scientific domain, as well as the role it has for the study programme. The course topics, the justification of including the course in the currcula of the study programme, etc. will be described in a general manner)*

The objective of the course is for the students to understand the concepts regarding the performance computing architectures used in the field of information technology. The course presents synthetically the languages used in information processing platforms such as C/C++/Java/Python/.Net and continues with parallel computing: multicore CPU and GPU architectures. The second part of the course deepens the GPU architecture and more specifically the CUDA architecture as a high-performance computing platform.

The laboratory applications aim to highlight the practical concepts and notions specific to calculation and information processing. Using the development of CUDA applications as methods of experimentation, or using technologies such as OpenCL, students can themselves and experience the knowledge acquired during the course. The whole set of notions acquired during the course can be experienced during laboratory applications

7. Competences *(Proven capacity to use knowledge, aptitudes and personal, social and/or methodological abilities in work or study situations and for personal and proffesional growth. They refflect the empolyers requirements.)*

Specific Competences	After completing this course, students acquire the basic and advanced notions of programming using GPU/CUDA architectures. This knowledge combined with going through laboratory applications gives the possibility to develop parallel information processing applications. Also, students will be able to develop techniques and methods for programming, optimization and efficiency of parallel algorithms.
Transversal (General) Competences	The fulfillment of professional tasks with the exact identification of the objectives to be achieved, potential risk factors, available resources, economic-financial aspects, the conditions for their completion, work stages, working time and related deadlines.

8. Learning outcomes *(Synthetic descriptions for what a student will be capable of doing or showing at the completion of a course. The learning outcomes reflect the student's acomplishments and to a lesser extent the teachers' intentions. The learning outcomes inform the students of what is expected from them with respect to performance and to obtain the desired grades and ECTS points. They are defined in concise terms, using verbs similar to the examples below and indicate what will be required for evaluation. The learning outcomes will be formulated so that the correlation with the competences defined in section 7 is highlighted.)*

Knowledge	*The result of knowledge aquisition through learning. The knowledge represents the totality of facts, priciples, theories and practices for a given work or study field. They can be theoretical and/or factual.* After completing this course, students acquire the basic and advanced notions of programming using parallel architectures such as multicore CPU and GPU/CUDA architectures. This knowledge combined with going through practical applications gives the possibility to develop efficient information processing applications. Also, students will be able to develop techniques and methods for programming, optimization and efficiency of information processing algorithms.
Skills	*The capacity to apply the knowledge and use the know-how for completing tasks and solving problems. The skills are described as being cognitive (requiring the use of logical, intuitive and creative thinking) or practical (implying manual dexterity and the use of methods, materials, tools and intrumentation).* Select and group relevant information in a given context.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* Demonstrates responsiveness to new learning contexts.

9. Teaching techniques *(Student centric techniques will be considered. The means for students to participate in defining their own study path, the identification of eventual fallbacks and the remedial measures that will be adopted in those cases will be described.)*

Within this subject, considering both the course hours and those of applications, both expository (lecture, exposition) and conversational-interactive teaching methods will be used, based on discovery learning models facilitated by the direct and indirect exploration of reality (experiment, demonstration, modelling), but also on action-based methods, such as exercise, practical activities and problem solving.

In the teaching activity, lectures will be used, based on some Power Point presentations. Presentations use images and diagrams so that the information presented is easy to understand and assimilate.

Presentations can also be made using online communication channels, on the Teams platform.

Active listening and assertive communication skills will be considered, as well as feedback construction mechanisms. Teamwork skills will be practiced to solve different learning tasks.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**
1	Introduction in parallel computing 1.1. Motivating parallelism: intensive-computing applications 1.2. Introduction of parallelism in computer architectures	2
2	GPU architectures 2.1. GPU, GPGPU architectures 2.2. CUDA architecture	8

3	CUDA architecture 3.1. History, evolution 3.2. Description 3.3. CUDA specific concepts 3.4. CUDA programming models 3.5. Programming patterns. Data level parallelism. CUDA memory types. 3.6. Applications. Analysis and optimization. Performance considerations.	10
4	CUDA based libraries 4.1. Presentation of libraries based on the CUDA architecture 4.2. cuBLAS, OpenACC, OpenCL, DNN	4
5	Applications 5.1. Applications of information processing with GPU/CUDA implementations: image processing, artificial intelligence, computer vision, 3D graphics	4
	Total:	28

Bibliography:
David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors 2nd edition - A Hands-on Approach, Morgan Kaufmann 2013
nVidia CUDA: www.developer.nvidia.com
Matt Pharr , Randima Fernando : GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Pearson Education 2005, http://download.nvidia.com/developer/GPU_Gems_2/CD/Index.html
Jason sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley 2010, http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf

LABORATORY

Crt. no.	Content	No. hours
1	GPU/CUDA libraries	2
2	Development of a CUDA application	2
3	Development of a CUDA application	2
4	Analysis/profiling CUDA applications	2
5	Optimization	2
6	Applications: image processing	2
7	Applications: data processing	2
	Total:	

Bibliography:
CUDA C Programming Guide, http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
CUDA C Best Practices Guide, http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html
David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors 2nd edition - A Hands-on Approach, Morgan Kaufmann 2013
Jason sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison Wesley 2010, http://www.mat.unimi.it/users/sansotte/cuda/CUDA_by_Example.pdf

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	- knowledge of fundamental theoretical terms; - knowledge of programming techniques and architectures; - knowledge of the GPU/CUDA platform; - knowledge of a library based on GPU/CUDA	Verification tests during the semester, held on fixed dates at the end of the course; the subjects cover the entire subject, realizing a synthesis between the comparative theoretical course of the subject and the explanation through exercises of the techniques specific to the programming of GPU platforms.	50%
11.5 Seminary/laboratory/project	- knowing how to design an image analysis algorithm to solve a given problem; - knowledge of how to transpose an algorithm into code; - demonstration of the operation of an implemented algorithm.	The final laboratory project, comprising a theoretical component and a practical component. The theoretical component is verified by preparing a report; the practical component is evaluated by checking the way of solving (implementation, testing, operation) by the student of some practical problems.	50%
11.6 Passing conditions			
Designing, implementing, and demonstrating of practical solutions for image processing, computer vision, artificial intelligence using CPU or GPU/CUDA accelerated libraries. Obtaining at least 50% of the total score. Obtaining at least 50% of the semester score.			

12. Corroborate the content of the course with the expectations of representatives of employers and representative professional associations in the field of the program, as well as with the current state of knowledge in the scientific field approached and practices in higher education institutions in the European Higher Education Area (EHEA)

Information processing requires knowledge of the architectures currently available on the market, the programming languages, the necessary data structures as well as the existing libraries adapted to various fields such as image processing, computer vision, artificial intelligence, 3D graphics, mathematical calculations, algebra. Understanding these aspects together with the particularities of each hardware architecture creates the prerequisites for efficient implementations, optimized and adapted to the ever-increasing computing requirements with the increasing complexity of applications and their data. The rapid processing of information requires the use of adapted platforms, whether cloud architectures, GPU architectures or mobile platforms are used.

Universitatea Națională de Știință și Tehnologie Politehnica București

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Date	Course lecturer	Instructor(s) for practical activities
15.09.2025	S.l./Lect. Dr. George Valentin STOICA	S.l./Lect. Dr. George Valentin STOICA

Date of department approval	Head of department
22.10.2025	Prof. Dr. Claudius Dan

Date of approval in the Faculty Council	Dean
	Prof. dr. ing. Radu Mihnea UDREA