SYLLABUS

1. Information regarding the program

1.1 Higher education institution	**Babeş-Bolyai University Cluj-Napoca**
1.2 Faculty	**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 program/Qualification	**High Performance Computing**

2. Information regarding the discipline

2.1 Name of discipline	**General Purpose GPU Programming**						
2.2 Course coordinator	**Lect. Rareş Florin Boian PhD**						
2.3 Seminar coordinator	**Lect. Rareş Florin Boian PhD**						
2.4 Year of study	2	2.5 Semester	2	2.6. Type of evaluation	E	2.7 Type of discipline	Optional

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

3.1 Hours per week	3	Of which: 3.2 lecture	2	3.3 Seminar/laboratory	1
3.4 Total hours in the curriculum	42	Of which: 3.5 lecture	28	3.6 Seminar/laboratory	14
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					20
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					20
Tutorship					15
Evaluations					8
Other activities:					-

3.7 Total individual study hours	83
3.8 Total hours per semester	125
3.9 Number of ECTS credits	8

4. Prerequisites (if applicable)

4.1 Curriculum	•
4.2 Competencies	•

5. Conditions (if applicable)

5.1 For the course	• Course information, lecture, seminar and laboratory activities are available at the following URL http://www.cs.ubbcluj.ro/~rares/course/gpgpu
5.2 For the seminar/lab activities	• Authenticated access to UNIX and Windows machines in the laboratories • Course information, lecture, seminar and laboratory activities are available at the following URL o http://www.cs.ubbcluj.ro/~rares/course/gpgpu

6. Specific competencies acquired

Profession competencies	• Define the basic concepts, theories and models of general purpose GPU programming • Analyze critically and apply the principles, the methods and the techniques to evaluate quantitatively and qualitatively the processes of GPU-based applications • Apply the concepts and fundamental theories of GPU programming to elaborate professional projects.
Transversal competencies	• Accomplish requirements according to the specifications, within the set deadlines, and respecting the professional ethics • Continuous study and learning in the specific domain of study • Focus on a high level of quality of the results of professional work, through personal involvement in the professional activities

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

7.1 General objective of the discipline	Introducing the students to general purpose GPU programming. The students should learn the following concepts: parallel architectures, GPU architecture, memory organization, parallelizable algorithms. The students should also learn to use GPU programming APIs and apply them practically in projects
7.2 Specific objective of the discipline	• GPU architecture • Memory organization • Work scheduling • NVIDIA CUDA • OpenCL • Microsoft DirectCompute • Parallelizable algorithms

8. Content

8.1 Lecture	Teaching methods	Remarks
Weeks 1 - 5 1. Optimizing matrix multiplication 2. Flynn taxonomy; Examples and applications; GPU hardware architecture 3. Introduction to NVIDIA CUDA 4. Introduction to OpenCL 5. Introduction to Microsoft DirectCompute	Exposition: presentation explanation, practical examples, demonstrations and case studies.	
Weeks 6 - 7 6. Applications: Ray tracing introduction. Lighting calculation. 7. Applications: Shadows, reflections, refractions in ray tracing	Exposition: presentation explanation, practical examples, demonstrations and case studies.	
Weeks 8 - 10 8. GPU memory organization 9. GPU work scheduling: blocks, threads, atomic	Exposition: presentation explanation, practical	

operations, synchronization 10. Measuring performance	examples, demonstrations and case studies.	
Weeks 11 - 14 11. Applications: Fractals 12. Applications: Graph partitioning 13. Applications: Parallel sorting: bubble sort, merge sort, quick sort, bitonic sort 14. Applications: Volumetric ray tracing	Exposition: presentation explanation, practical examples, demonstrations and case studies.	

Bibliography

1. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Introduction to Parallel Computing, Addison-Wesley; 2 edition (January 26, 2003)
2. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional; 1 edition (July 29, 2010)
3. Nicholas Wilt, The CUDA Handbook: A Comprehensive Guide to GPU Programming, Addison-Wesley Professional; 1 edition (June 11, 2013)
4. Aaftab Munshi, Benedict Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg, OpenCL Programming Guide, Addison-Wesley Professional; 1 edition (July 23, 2011)
5. Matthew Scarpino, OpenCL in Action: How to Accelerate Graphics and Computations, Manning Publications (November 17, 2011)
6. Russ Miller, Laurence Boxer , Algorithms Sequential & Parallel: A Unified Approach, Cengage Learning; 3 edition (December 20, 2012)

8.2 Seminar / laboratory	Teaching methods	Remarks
1. CUDA programming examples	Explanation, examples, case studies, dialog	
2. OpenCL programming examples	Explanation, examples, case studies, dialog	
3. Microsoft DirectCompute programming examples	Explanation, examples, case studies, dialog	
4. Ray tracer design	Explanation, examples, case studies, dialog	
5. Ray tracer optimizations	Explanation, examples, case studies, dialog	
6. Semester project	Explanation, examples, case studies, dialog	
7. Project grading		

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

- Learning the theoretical and methodological concepts and the practical aspects included in the General Purpose GPU Programming course, students acquire the knowledge required by the Grid on – RNCIS for the partial competences
- The course follows the IEEE and ACM Curricula Recommendations for Computer Science studies.
- The course exists in the curricula of similar departments in Romania

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Lecture	Understand and be able to use the concepts and methods taught by the course	Written exam	40%
10.5 Seminar/laboratory	Ability to correctly solve practical problems from the course curricula	Semester project	60%
10.6 Minimum performance standards			
• The students must get a grade of at least 5 for each of the two evaluations			

Date Signature of course coordinator Signature of seminar coordinator

Lect. Rareş Florin Boian PhD Lect. Rareş Florin Boian PhD

Date of departmental approval Signature of the department chair

..... Prof. Bazil Pârv PhD