# SYLLABUS

*GPU and distributed architecture computing*

University year 2025-2026

**1. Information regarding the programme**

| | | |
|---|---|---|
| 1.1. Higher education institution | Babes-Bolyai University | |
| 1.2. Faculty | Mathematics and Computer Science | |
| 1.3. Department | Computer Science | |
| 1.4. Field of study | Computer Science | |
| 1.5. Study cycle | Master | |
| 1.6. Study programme/Qualification | High Performance Computing and Big Data Analytics | |
| 1.7. Form of education | Full time | |

**2. Information regarding the discipline**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1. Name of the discipline | **High Performance Computing and Big Data Analytics** | | | | | Discipline code | **MME8111** |
| 2.2. Course coordinator | | | | Assoc. prof. Rareş Florin Boian, Ph.D. | | | |
| 2.3. Seminar coordinator | | | | Assoc. prof. Rareş Florin Boian, Ph.D. | | | |
| 2.4. Year of study | 2 | 2.5. Semester | 3 | 2.6. Type of evaluation | E | 2.7. Discipline regime | Mandatory |

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

| | | | | | |
|---|---|---|---|---|---|
| 3.1. Hours per week | **4** | of which: 3.2 course | **2** | 3.3 seminar/laboratory/project | **2** |
| 3.4. Total hours in the curriculum | 56 | of which: 3.5 course | 28 | 3.6 seminar/laboratory/project | **28** |

| **Time allotment for individual study (ID) and self-study activities (SA)** | **hours** |
|---|---|
| Learning using manual, course support, bibliography, course notes (SA) | 25 |
| Additional documentation (in libraries, on electronic platforms, field documentation) | 25 |
| Preparation for seminars/labs, homework, papers, portfolios and essays | 25 |
| Tutorship | 7 |
| Evaluations | 6 |
| Other activities: | |

| | |
|---|---|
| **3.7. Total individual study hours** | **88** |
| **3.8. Total hours per semester** | **200** |
| **3.9. Number of ECTS credits** | **8** |

**4. Prerequisites** (if necessary)

| | |
|---|---|
| 4.1. curriculum | |
| 4.2. competencies | |

**5. Conditions** (if necessary)

| | |
|---|---|
| 5.1. for the course | |
| 5.2. for the seminar /lab activities | |

**6.1. Specific competencies acquired** [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

| | |
|---|---|
| **Professional/essential competencies** | • advanced programming skills in high-level programming languages<br>• development and maintenance of software systems<br>• use of software tools in an interdisciplinary context |
| **Transversal competencies** | • application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms<br>• use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language |

**6.2. Learning outcomes**

| | |
|---|---|
| **Knowledge** | • Absolventul are cunoştinţe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor şi aplicaţiilor software, procesarea informaţiilor.<br>• Absolventul are cunoştinţe necesare pentru revizuirea literaturii de specialitate. |
| **Skills** | • Absolventul are abilitatea de a dezvolta, proiecta şi crea noi aplicaţii, sisteme sau produse folosind bunele practici din domeniu.<br>• Absolventul are aptitudinile necesare pentru conceperea programelor de calculator şi analiza sistemelor software.<br>• Absolventul are deprinderile necesare pentru utilizarea instrumentelor de sprijinire a cercetării. |
| **Responsibility and autonomy:** | • Absolventul este capabil să identifice probleme complexe şi să examineze probleme conexe pentru a dezvolta opţiuni de rezolvare şi implementa soluţii.<br>• Absolventul este capabil să combine informaţii diverse pentru a formula soluţii şi genera idei de dezvoltare pentru noi produse şi aplicaţii.<br>• Absolventul este capabil sa aplice şabloane arhitecturale, şabloane de proiectare si bunele practici in domeniu pentru a proiecta aplicaţii software de complexitate mare.<br>• Absolventul are capacitatea de a evalua diferite arhitecturi si soluţii posibile pentru o problema si a alege pe cel potrivit pentru cerinţele si constrângerile specifice aplicaţiei de dezvoltat. |

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

| | |
|---|---|
| **7.1 General objective of the discipline** | • Introduction to GPU programming<br>• Learning advanced concepts on GPU and distributed architecture computing<br>• Distributed architectures |
| **7.2 Specific objective of the discipline** | • GPU<br>  ○ GPU architecture<br>  ○ Memory organization<br>  ○ Work scheduling<br>  ○ NVIDIA CUDA<br>  ○ OpenCL<br>  ○ Parallelizable algorithms<br>• Distributed programming paradigms |

**8. Content**

| 8.1 Course | Teaching methods | Remarks |
|---|---|---|
| Weeks 1-3 Introduction to CUDA programming | Exposition: presentation explanation, practical examples, demonstrations and case studies. | |
| Weeks 4-7 CUDA architecture and memory types | Exposition: presentation explanation, practical examples, demonstrations and case studies. | |
| Weeks 8-9 Advanced CUDA functions | Exposition: presentation explanation, practical examples, demonstrations and case studies. | |
| Weeks 10-12 Distributed architectures | Exposition: presentation explanation, practical examples, demonstrations and case studies. | |
| Weeks 13-14 Distributed programming paradigms | Exposition: presentation explanation, practical examples, demonstrations and case studies. | |

Bibliography


1. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010
2. Gerassimos Barlas, Multicore and GPU Programming: An Integrated Approach, Morgan Kaufmann, 2022
3. Tolga Soyata, GPU Parallel Program Development Using CUDA, Chapman and Hall/CRC, 2018
4. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, A Hands-on Approach, 2ed, Morgan Kaufmann, 2010
5. Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Media, 2017
6. Maarten van Steen, Andrew S. Tanenbaum, Distributed Systems, Pearson Education, 2017
7. Paul Butcher, Seven Concurrency Models in Seven Weeks, Pragmatic Bookshelf, 2014
8. W. Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, 3ed, Addison-Wesley Professional, 2013


| 8.2 Seminar / laboratory | Teaching methods | Remarks |
|---|---|---|
| GPU project introduction | Explanation, examples, case studies, dialog | |
| GPU Project implementation | Explanation, examples, case studies, dialog | |
| GPU Project implementation | Explanation, examples, case studies, dialog | |
| Distributed project introduction | Explanation, examples, case studies, dialog | |
| Distributed project implementation | Explanation, examples, case studies, dialog | |
| Distributed project implementation | Explanation, examples, case studies, dialog | |
| Project grading | Discussions | |

Bibliography


1. Jason Sanders, Edward Kandrot, CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010
2. W. Stevens, Stephen Rago, Advanced Programming in the UNIX Environment, 3ed, Addison-Wesley Professional, 2013


**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 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**

| Activity type | 10.1 Evaluation criteria | 10.2 Evaluation methods | 10.3 Percentage of final grade |
|---|---|---|---|
| 10.4 Course | The level of knowledge and understanding of the course subjects | Written exam | 30% |
| 10.5 Seminar/laboratory | Ability to solve practical problems, specific to the course subjects | CUDA project | 30% |
| | | Distributed project | 30% |
| 10.6 Minimum standard of performance | | | |
| • Minimum 5 in the final grade | | | |

**11. Labels ODD (Sustainable Development Goals)[2]**

*Not applicable*.

Date:
...

Signature of course coordinator

.....

Signature of seminar coordinator

.....

Date of approval:
...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable*.".