

FIȘA DISCIPLINEI

Programare pe arhitecturi GPU si distribuite

Anul universitar 2025-2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Facultatea	Matematică și Informatică
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Master
1.6. Programul de studii / Calificarea	Calcul de înaltă performanță și analiza volumelor mari de date
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare pe arhitecturi GPU si distribuite				Codul disciplinei		MME8111
2.2. Titularul activităților de curs			Conf. dr. Rareș Boian					
2.3. Titularul activităților de seminar			Conf. dr. Rareș Boian					
2.4. Anul de studiu	2	2.5. Semestrul	3	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligativu	

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2
3.4. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	28
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri (mai mare sau egal cu nr. total ore prevăzut în calendarul disciplinei pentru temele de control)					25
Tutoriat (consiliere profesională)					7
Examinări					6
Alte activități [de ex.: comunicare bidirecțională cu titularul de disciplină / tutorele]					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				88	
3.8. Total ore pe semestru				200	
3.9. Numărul de credite				8	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	
4.2. de competențe	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	
5.2. de desfășurare a seminarului/ laboratorului	

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/ esențiale	• programarea în limbaje de nivel înalt • dezvoltarea și întreținerea aplicațiilor informatice • utilizarea instrumentelor informatice în context interdisciplinar
Competențe transversale	• aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională • utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională

6.2. Rezultatele învățării

Cunoștințe	• Absolventul are cunoștințe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor. • Absolventul are cunoștințe necesare pentru revizuirea literaturii de specialitate.
Aptitudini	• Absolventul are abilitatea de a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu. • Absolventul are aptitudinile necesare pentru conceperea programelor de calculator și analiza sistemelor software. • Absolventul are deprinderile necesare pentru utilizarea instrumentelor de sprijinire a cercetării.
Responsabilități și autonomie	• Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții. • Absolventul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații. • Absolventul este capabil să aplice șabloane arhitecturale, șabloane de proiectare și bunele practici în domeniu pentru a proiecta aplicații software de complexitate mare. • Absolventul are capacitatea de a evalua diferite arhitecturi și soluții posibile pentru o problemă și a alege pe cel potrivit pentru cerințele și constrângerile specifice aplicației de dezvoltat.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Introducere în programarea GPU • Însușirea conceptelor avansate de programare GPU și distribuită • Arhitecturi distribuite
7.2 Obiectivele specifice	• GPU ○ Arhitectura GPU ○ Organizarea memoriei ○ Planificarea sarcinilor ○ NVIDIA CUDA ○ OpenCL ○ Algoritmi paralelizabili • Paradigme de programare distribuita

8. Conținuturi

8.1 Curs	Metode de predare	Observații
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.	
Bibliografie 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 / laborator	Metode de predare	Observații
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	
Bibliografie 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. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Prin însușirea conceptelor teoretico-metodologice și abordarea aspectelor practice incluse în disciplina Algoritmi, modele și concepte în sisteme distribuite, studenții dobândesc un bagaj de cunoștințe consistent, în concordanță cu competențele parțiale cerute pentru ocupațiile posibile prevăzute în Grila 1 – RNCIS
- Cursul respectă IEEE and ACM Curricula Recommendations for Computer Science studies.
- Cursul există în programa de studii a universităților și facultăților de profil din România
- Conținutul cursului este foarte bine apreciat de către companiile de software care are ca și angajați absolvenți ai acestui curs

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Nivelul de cunoștințe și înțelegerea subiectelor cursului	Examen scris	30%
10.5 Seminar/laborator	Abilitatea de a rezolva probleme practice specifice subiectului cursului	Proiect CUDA	30%
		Proiect distribuit	30%
10.6 Standard minim de performanță			
Minimum 5 la nota finală			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)²

Nu se aplică.

Data completării:

...

Semnătura titularului de curs

.....

Semnătura titularului de seminar

.....

Data avizării în departament:

...

Semnătura directorului de departament

Conf.dr. Adrian STERCA

² Păstrați doar etichetele care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru Dezvoltare durabilă - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "Nu se aplică".