

FIȘA DISCIPLINEI

Programare paralelă și distribuită

Anul universitar 2026-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Facultatea	Facultatea de Matematica și Informatica
1.3. Departamentul	Departamentul de Informatica
1.4. Domeniul de studii	Informatica
1.5. Ciclu de studii	Licenta
1.6. Programul de studii / Calificarea	Informatica
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Programare paralelă și distribuită			Codul disciplinei	MLE5077
2.2. Titularul activităților de curs	Conf. univ.dr. Niculescu Virginia				
2.3. Titularul activităților de seminar	Conf. univ.dr. Niculescu Virginia				
2.4. Anul de studiu	3	2.5. Semestrul	5	2.6. Tipul de evaluare	Examen
2.7. Regimul disciplinei	Obligativu		2.8. Tipul disciplinei	Disciplină fundamentală (DF)	

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/ proiect	3
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					5
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					30
Tutoriat (consiliere profesională)					5
Examinări					5
Alte activități					0
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				55	
3.8. Total ore pe semestru				125	
3.9. Numărul de credite				5	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Programare orientată obiect; Sisteme de operare Metode avansate de programare Medii de proiectare și programare
4.2. de competențe	Abilități de programare Java și C++

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sala cu proiector
5.2. de desfășurare a seminarului/ laboratorului	Laborator cu stații de lucru

6.1. Competențele dobândite în urma absolvirii programului de studii (se preiau din planul de învățământ)¹

¹ Se vor prelua din Planul de învățământ al programului de studii acele competențe profesionale și/sau transversale la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa disciplinei. Pentru fiecare competență se va prelua întregul enunț, inclusiv codul competenței, cu formularea care apare în planul de

Competențe profesionale	
Codul competenței	Competență
CP1	crează softuri
CP5	defineste cerinte tehnice
CP10	utilizează biblioteci de software
CP11	utilizeaza sabloane de proiectare de software
Competențe transversale	
Codul competenței	Competență
CT1	Lucrează independent
CT2	Solucioneaza probleme
CT3	Gândește analitic

6.2. Rezultatele învățării specifice programului de studii (se preiau din planul de învățământ)²

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP1 CP6	1. Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și frameworkuri actuale	1. Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software scalabile și utilizează eficient resursele hardware și software.
CP9 CP10	2. Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și frameworkuri actuale.	2. Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software scalabile și utilizează eficient resursele hardware și software.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Studentul/absolventul intelege problemele ridicate de aplicatiile paralele si distribuite
2. Studentul/absolventul cunoaste mecanismele de baza si framework-urile utilizate in constructia aplicatiilor paralele si distribuite
3. Studentul/absolventul cunoaste pattern-urile utilizate in dezvoltarea aplicatiilor paralele si distribuite
Abilități academice specifice (Specific academic skills)
1. Studentul/absolventul este capabil sa proiecteze algoritmi si aplicatii paralele si distribuite
2. Studentul/absolventul intelege problemele ridicate de paralelizare si e capabil sa gaseasca solutii la acestea
3. Studentul/absolventul e capabil sa evalueze si sa determine imbunatatirile de performanta obtinute prin paralelizare.

8. Conținuturi

învățământ, fără modificări. Dacă nu se preia nici o competență din oricare din cele două categorii, se șterge linia din tabel aferentă acelei categorii.

² Se menționează rezultatele învățării specifice programului de studiu la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa. Enunțurile, preluate fără modificări din Planul de învățământ în funcție de tipul disciplinei (DF/DS/DC) se trec în dreptul competenței asociate.

8.1 Curs	Metode de predare - învățare	Observații³
C1 Introducere generala in programarea paralela si distribuita	Expunere	
C2. Arhitecturi paralele – <i>Taxonomii (Flynn)</i>	Expunere	
C3. Tipuri si modele de paralelism - Modelul Data-parallel - Modelul Message-passing (distributed memory) - Modelul Shared-memory Procese versus fire de executie.	Expuneri: concepte, exemple, studii de caz	
C4. Programare paralela in medii cu memorie distribuita. <i>MPI</i>	Expuneri: concepte, exemple, studii de caz	
C5. Programare paralela in medii cu memorie partajata	Expuneri: concepte, exemple, studii de caz	
C6. Concurenta – concepte Sincronizare: monitoare, semafoare, variabile conditionale	Expuneri: concepte, exemple, studii de caz	
C7. Concurenta – patterns. Producator-Consumator	Expuneri: concepte, exemple, studii de caz	
C8. Programare paralela in medii cu memorie partajata. - OpenMP	Expuneri: concepte, exemple, studii de caz	
C8. Programare paralela asincrona	Expuneri: concepte, exemple, studii de caz	
C9 . Evaluarea performantei programelor paralele: complexitate-timp, complexitate-procesor, acceleratie, eficienta, cost; granularitate, scalabilitate	Expuneri: concepte, exemple, studii de caz	
C10. <i>Data parallel programming</i> : Cadrul general de dezvoltare a aplicațiilor GPU Modelul de programare CUDA	Expuneri: concepte, exemple, studii de caz	
C11- Sabloane de proiectare pentru programarea paralela <i>Master-slaves; Task-Farm/Work-Pool; Divide & Impera;</i>	Expuneri: concepte, exemple, studii de caz	
C12- Sabloane de proiectare pentru programarea paralela <i>Divide & Impera; Pipeline</i>	Expuneri: concepte, exemple, studii de caz	
C13 Sabloane de proiectare pentru programarea distribuita	Expuneri: concepte, exemple, studii de caz	
C14. Analiza comparativa generala a noilor concepte/tehnici/principii/sabloane introduse	Analiza, exemple, studii de caz	
Bibliografie		
<ol style="list-style-type: none"> 1. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995. 2. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012. 3. F. Buschmann, K. Henney, D. C. Schmidt. Pattern-Oriented Software Architecture Volume 4: A Pattern Language for 4. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003. 5. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000. 6. B. L. Massingill, T.G. Mattson, and B. A. Sanders, A Pattern Language for Parallel Programming. Wesley Software Patterns Series, 2004. 7. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006. 8. M. Richards. Software Architecture Patterns. Understanding Common Architecture. Patterns and When to Use Them 2015 O'Reilly Media. 9. D. Schmidt, M. , H. Rohnert, F. Buschmann. Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects Volume 2. Wiley, 2000. 10. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002 <p>***, Tutoriale OpenMP ***, Tutoriale MPI ***, Tutoriale CUDA</p>		
8.2 Seminar / laborator	Metode de predare - învățare	Observații



³ De exemplu aspecte organizatorice, recomandări pentru studenți, aspecte specifice legate de curs/seminar cum ar fi invitarea unor practicieni în domeniu etc.

L1 –L4 . Programare multithreading C++ si Java	Discutii, exemplificare, evaluare	
L5-L6 Programare distribuita MPI	Discutii, exemplificare, evaluare	
L7- L9. Concurenta – conditionari, sincronizari	Discutii, exemplificare, evaluare	
L10. Programare OpenMP	Discutii, exemplificare, evaluare	
L11, Proiect CUDA	Discutii, exemplificare, evaluare	
L12- L14 Proiect complex Client-Server	Discutii, exemplificare, evaluare	
Bibliografie <ol style="list-style-type: none"> 1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006. 2. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002. 3. J. Sanders, E. Kandrot. CUDA by Example. An Introduction to General-Purpose GPU Programming. Addison-Wesley. 2010. 4. A. WILLIAMS. C++ Concurrency in Action. PRACTICAL MULTITHREADING. MANNING, 2012. 5. ***, Tutoriale Java http://download.oracle.com/javase/tutorial/ 6. ***, Tutorial C++ [https://en.cppreference.com/] 7. ***, OpenMP[http://openmp.org/wp/] 8. ***, MPI[http://www.mpi-forum.org/] 		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ⁴	9.2 Metode de evaluare ⁵	9.3 Pondere din nota finală
9.4 Curs	Cunoasterea conceptelor si mecanismelor de baza	Examen teoretic	40%
9.5 Seminar/laborator	Folosirea conceptelor si mecanismelor introduse la curs pentru rezolvarea unor probleme concrete	Programe de laborator - Evaluarea calitatii si completitudinii temelor primite -Evaluarea intelegerii aplicarii conceptelor si mecanismelor folosite	40%
	Proiect complex	Prezentare, verificare a implementarii si a documentatiei asociata	20%
9.6 Standard minim de promovare			
<ul style="list-style-type: none"> o Media finala trebuie sa fie cel putin 5 (pe o scara de la 1 la 10). o Promovarea este condiționată de nota minimă 5 la examenul teoretic 			

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)⁶

		Eticheta generală pentru Dezvoltare durabilă
---	---	--

⁴ Criteriile de evaluare trebuie să reflecte direct rezultatele învățării vizate la nivel de program de studii, respectiv la nivel de disciplină. Mai concret, se evaluează achizițiile de învățare menționate în rezultatele anticipate ale învățării.

⁵ Se recomandă stabilirea atât a metodelor de evaluare finală, cât și a strategiei de evaluare pe parcurs.

⁶ Selectați o singură etichetă, cea care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivește cel mai bine disciplinei. Dacă disciplina tratează tema dezvoltării durabile la modul general (de ex. prin prezentarea/introducerea cadrului general al dezvoltării durabile etc.) atunci se poate alocă eticheta generală de Dezvoltare Durabilă. Dacă niciuna dintre etichete nu descrie disciplina, selectați ultima opțiune: „Nu se aplică nici o etichetă”.

								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	X
								Nu se aplică nici o etichetă
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Data completării:
10.05.2026

Semnătura titularului de curs
Virginia Niculescu

Semnătura titularului de seminar
Virginia Niculescu

Data avizării în departament:

...

Semnătura directorului de departament

.....