

FIȘA DISCIPLINEI
Fundamentele Programării
Anul universitar 2026-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Fundamentele programării			Codul disciplinei	MLR5005
2.2. Titularul activităților de curs	Prof. dr. CZIBULA Istvan Gergely				
2.3. Titularul activităților de seminar	Prof. dr. CZIBULA Istvan Gergely				
2.4. Anul de studiu	1	2.5. Semestrul	1	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	6	din care: 3.2. curs	2	3.3. seminar/ laborator/proiect	2 sem 2 lab
3.4. Total ore din planul de învățământ	82	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	56
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					14
Tutoriat (consiliere profesională)					8
Examinări					18
Alte activități					
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				66	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

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	Sală, plus proiector
5.2. de desfășurare a seminarului/ laboratorului	Laboratoare echipate cu Python

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

Competențe profesionale	
Codul competenței	Competență
CP1	Creează softuri
CP4	Definește arhitectura software
CP5	Definește cerințe tehnice
Competențe transversale	
Codul competenței	Competență
CT2	Soluționează 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	1. Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.	1. Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
CP4	2. Studentul/absolventul cunoaște și explică paradigmele moderne de programare, arhitecturi software și metodologii de dezvoltarea proiectelor software.	2. Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software scalabile și utilizează eficient resursele hardware și software.
CP5	3. Studentul/absolventul alege, explică și specifică fundamentele matematice aplicate în informatică, inclusiv logica formală, algebra, probabilitățile și statisticile.	3. Studentul/absolventul aplică, evaluează, propune metodele matematice pentru modelarea, simularea și rezolvarea problemelor informatice.
CT2	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții.
CT3	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul este capabil să combine informații diverse pentru a formula soluții și genera idei de dezvoltare pentru noi produse și aplicații.

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 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.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Studentul cunoaște conceptele de bază ale ingineriei software (proiectare, implementare și întreținere).
2. Studentul cunoaște limbajul de programare Python.
2. Studentul înțelege conceptele de bază ale programării.
Abilități academice specifice (Specific academic skills)
1. Studentul este capabil să folosească instrumente de bază pentru construirea programelor.
2. Studentul este capabil să programeze în limbajul Python și să folosească instrumente de dezvoltare pentru programarea, execuția și depanarea programelor Python.
3. Studentul are un stil de programare conform celor mai bune recomandări practice.

8. Conținuturi

8.1 Curs	Metode de predare - învățare	Observații³
1. Introducere în procese de dezvoltare software <ul style="list-style-type: none"> Ce este programarea: algoritm, program, elemente de bază Python, interpretor Python, roluri în ingineria software Cum scriem programe: enunț problemă, cerințe, proces de dezvoltare dirijat de funcționalități (FDD) Exemple: calculator 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
2. Programare procedurală <ul style="list-style-type: none"> Tipuri structurate: liste, tuple, dicționare Funcții: cazuri de testare, definire, variabile, apel Transmiterea parametrilor Funcții anonime Cum scriem funcții: programare dirijată de teste, refactorizări 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
3. Programare modulară <ul style="list-style-type: none"> Ce este un modul: modul Python, domeniul variabilelor, pachete, module standard, distribuie module Cum organizăm codul sursă: responsabilități, single responsibility principle, separation of concerns, dependency, coupling, cohesion Arhitecturi software stratificate Eclipse+PyDev 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
4. Tipuri definite de utilizator <ul style="list-style-type: none"> Cum definim tipuri noi Încapsulare, ascunderea informației, tipuri abstracte de date 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	
5. Principii de proiectare și programare <ul style="list-style-type: none"> Problema: program cu operații CRUD pe entități de un tip dat Arhitectura stratificată: UI, Domeniu, Infrastructura 	<ul style="list-style-type: none"> Expunerea interactivă Explicația Conversația Demonstrația didactică 	

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

<ul style="list-style-type: none"> • Şabloane GRASP • Şabloane DDD: entity, validator, repository, controller • Principii: Information Expert, Low Coupling, High Cohesion, Protected Variation, Single responsibility, Dependency Injection 		
6. Programare orientată pe obiecte <ul style="list-style-type: none"> • Obiecte și clase • Diagrame UML • Moștenire • Excepții 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
7. Proiectarea programelor <ul style="list-style-type: none"> • Top down and bottom up strategies • Organizarea elementelor UI și relația cu alte straturi 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
8. Testarea și inspectarea programelor <ul style="list-style-type: none"> • Black box testing, white box testing • Unit testing, integration testing • Program inspection: coding style, refactoring 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
9. Recursivitate <ul style="list-style-type: none"> • Recursivitate directă și indirectă • Exemple Complexitatea algoritmilor <ul style="list-style-type: none"> • Notăția asimptotică: big-o, little-o, big-omega, little-omega, theta • Comparații algoritmi 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
10. Algoritmi de căutare <ul style="list-style-type: none"> • căutare secvențială • căutare binară Algoritmi de sortare <ul style="list-style-type: none"> • BubbleSort • SelectionSort • InsertionSort • QuickSort • MergeSort • Complexitatea algoritmilor 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
11. Backtracking <ul style="list-style-type: none"> • Algoritmul Backtracking • Extensii ale algoritmului • Exemple 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
12. Divide et Impera <ul style="list-style-type: none"> • Descriere Metodă • Exemple Greedy <ul style="list-style-type: none"> • Descriere Metoda Exemple	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
13 Programare dinamică <ul style="list-style-type: none"> • Descriere Metodă • Exemple 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
14. Recapitulare	<ul style="list-style-type: none"> • Expunerea interactivă • Conversația 	
Bibliografie		

1. Kent Beck. Test Driven Development: By Example. Addison-Wesley Longman, 2002. See also Test-driven development. http://en.wikipedia.org/wiki/Test-driven_development
2. Martin Fowler. Refactoring. Improving the Design of Existing Code. Addison-Wesley, 1999. See also <http://refactoring.com/catalog/index.html>
3. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006
4. The Python language reference. <http://docs.python.org/py3k/reference/index.html>
5. The Python standard library. <http://docs.python.org/py3k/library/index.html>
6. The Python tutorial. <http://docs.python.org/tutorial/index.html>

8.2 Seminar/laborator	Metode de predare - învățare	Observații
1. Programe Python	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
2. Programare procedurala	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
3. Programare modulara	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
4. Tipuri definite de utilizator	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
5. Principii de proiectare	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
6. POO	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
7. Proiectare	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
8. Testare si inspectare	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
9. Recursivitate	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie 	

	<ul style="list-style-type: none"> • Exemple • Demonstratie didactica 	
10. Complexitatea algoritmilor	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
11. Backtracking	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
12. Metoda injumatatirii. Algoritmi de cautare	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
13. Pregatirea examenului practic	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
14: Pregatirea examenului scris	<ul style="list-style-type: none"> • Expunere interactiva • Explicatie • Conversatie • Exemple • Demonstratie didactica 	
Bibliografie		
<ol style="list-style-type: none"> 1. Kent Beck. Test Driven Development: By Example. Addison-Wesley Longman, 2002. See also Test-driven development. http://en.wikipedia.org/wiki/Test-driven_development 2. Martin Fowler. Refactoring. Improving the Design of Existing Code. Addison-Wesley, 1999. See also http://refactoring.com/catalog/index.html 3. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006 4. The Python language reference. http://docs.python.org/py3k/reference/index.html 5. The Python standard library. http://docs.python.org/py3k/library/index.html 6. The Python tutorial. http://docs.python.org/tutorial/index.html 		

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	<ul style="list-style-type: none"> • Cunoștințele acumulate 	Examen scris	40%
9.5 Seminar/laborator	<ul style="list-style-type: none"> • Scrierea unui program 	Examen practic	30%
	<ul style="list-style-type: none"> • Programele scrise in timpul semestrului 	Documentatie	30%




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.

9.6 Standard minim de promovare

- Fiecare student trebuie să demonstreze că a atins un nivel acceptabil de cunoștințe și înțelegere a domeniului, că este capabil să prezinte aceste cunoștințe într-o manieră coerentă și că are abilitatea de a stabili anumite conexiuni și de a folosi aceste cunoștințe în rezolvarea diferitelor probleme în limbajul de programare Python.
- Pentru promovare, este obligatorie prezența la cel puțin 10 seminarii și 12 laboratoare .
- Promovarea este condiționată de nota minimă 5 la activitatea de laborator, proba practică și examenul scris.

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

	Eticheta generală pentru Dezvoltare durabilă							
								
<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 checked="" type="radio"/>
								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:

06.07.2026

Semnătura titularului de curs

Prof. dr. CZIBULA Istvan Gergely

Semnătura titularului de seminar

Prof. dr. CZIBULA Istvan Gergely

Data avizării în departament:

Semnătura directorului de departament

Conf. dr. Adrian STERCA

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”.