FIȘA DISCIPLINEI

Fundamentele Programării

Anul universitar 2025 - 2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai din 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	**MLE5005**
2.2. Titularul activităților de curs	Conf. Univ. Dr. Molnar Arthur-Jozsef						
2.3. Titularul activităților de seminar	Conf. Univ. Dr. Molnar Arthur-Jozsef						
2.4. Anul de studiu	**1**	2.5. Semestrul	**1**	2.6. Tipul de evaluare	**E**	2.7. Regimul disciplinei	**Obligatoriu**

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	**4**
3.4. Total ore din planul de învățământ	**84**	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)					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					20
Tutoriat (consiliere profesională)					5
Examinări					0
Alte activități					6
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 de curs cu video-proiector și acces Internet
5.2. de desfășurare a seminarului/ laboratorului	Sală de curs cu video-proiector, acces Internet și stații de lucru configurate pentru dezvoltarea software folosind limbajul Python.

6.1. Competențele specifice acumulate

Competențe profesionale/esențiale	• Programarea în limbaje de nivel înalt • Utilizarea bazelor teoretice ale informaticii și a modelelor formale
Competențe transversale	• 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. • 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.

6.2. Rezultatele învățării

Cunoștințe	• Absolventul are cunoștințe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor și aplicațiilor software, procesarea informațiilor. • Absolventul are abilitatea de a aplica reguli generale unor probleme specifice și de a produce soluții relevante.
Aptitudini	• Absolventul are aptitudinile necesare pentru înțelegerea și folosirea conceptelor programării orientate obiect la dezvoltarea unor aplicații software de complexitate medie. • Absolventul are aptitudinile necesare pentru conceperea programelor de calculator și analiza sistemelor software.
Responsabilități și autonomie	• Absolventul are capacitatea de a observa și obține informații din diverse surse. • Absolventul cunoaște aspectele de bază legate de gestiunea softului.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• Să cunoască conceptele de bază ale ingineriei software (proiectare, implementare și mentenanță) și să fie capabil de a le aplica cu scopul de a crea aplicații de dimensiune mică și medie în limbajul Python folosind paradigma programării structurate sau orientate obiect și care utilizează interfața cu utilizatorul în consolă.
7.2 Obiectivele specifice	• Să cunoască conceptele cheie ale programării. • Să cunoască conceptele de bază ale ingineriei software în domeniul proiectării, implementării și mentenanței sistemelor software. • Să capete o bună înțelegere asupra uneltelor software de bază utilizate în procesul de dezvoltare și testare. • Să cunoască limbajul Python și să dobândească competențele necesare executării, testării și depanării programelor Python. • Să dobândească și îmbunătățească propriul stil de programare.

8. Conținuturi

8.1 Curs	Metode de predare
Introducere în curs Ce este programarea: algoritm, program, elementele de bază ale limbajului Python, interpretorul Python.	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică
Recursivitatea. Complexitatea algoritmilor • Noțiunea de recursivitate, recursivitate directă și indirectă, exemple. • Analiză empirică și asimptotică. • Notația asimptotică: big-o, little-o, big-omega, little-omega, theta. • Compararea algoritmilor din punct de vedere al eficienței, analizarea atât a complexității timp cât și a complexității spațiului de memorie.	
Căutare. Sortare • Specificarea problemelor de căutare și sortare. • Metode de căutare: secvențială, binară, exponențială. • Metode de sortare: a bulelor, prin selecție, prin inserție (și optimizări), quick sort, prin interclasare (și optimizări). • Complexitatea timp și a spațiului de memorie pentru algoritmii de căutare și sortare.	
Metode de rezolvare a problemelor • Prezentarea metodelor greedy, divide et impera, backtracking și a programării dinamice. • Probleme tipice acestor metode. • Analizarea complexității timp și a spațiului de memorie.	
Programarea procedurală • Tipuri compuse în Python: listă, tuplă, dicționar. • Funcții: definiție, crearea cazurilor de testare, domeniu de vizibilitate al variabilelor, apel, transmiterea parametrilor. • Pașii metodologiei de dezvoltare bazate pe teste și refactorizarea.	
Programarea modulară • Ce este un modul, ce este un modul în Python, domeniul de vizibilitate al variabilelor într-un modul, pachete, biblioteci standard de module, instalarea modulelor adiționale. • Organizarea codului sursă: responsabilități, principiului singurei responsabilități, separarea responsabilităților, dependența, cuplarea, coeziunea. • Introducere în șablonul arhitecturii stratificate.	
Clase și obiecte • Definirea tipurilor de date adiționale în Python prin utilizarea claselor. • Metodele claselor, metode speciale, crearea obiectelor, vizibilitatea claselor și a pachetelor Python.	
Programarea bazată pe obiecte • Ascunderea informației, încapsularea, moștenirea, tipizarea dinamică. • Aplicarea principiilor arhitecturii stratificate folosind clase și obiecte. • Utilizarea mecanismului de excepții. • Exemple focalizate pe soluții bazate pe arhitectura stratificată ce demonstrează utilizarea ierarhiilor de clase, a fișierelor text și binare în scopul stocării datelor precum și utilizarea excepțiilor Python pentru a semnala apariția unei situații neașteptate.	
Principii pentru proiectarea programelor • Principii de proiectare: expertul în informație, cuplarea scăzută, coeziunea înaltă, principiul singurei responsabilități, injectarea dependențelor. • Limbajul Unified Modelling Language, diagramele de clase UML.	
Testarea și refactorizarea programelor • Introducere în testarea programelor, niveluri de testare al programelor. • Testarea manuală și automată folosind PyUnit. • Introducere în paradigma dezvoltării ghidate de testare. • Inspecția și refactorizarea programelor.	
Pregătirea examenului • Recapitularea celor mai importante teme acoperite în cadrul cursului. • Prezentarea ghidului de examen.	
Bibliografie	

1. Kent Beck - ***Test Driven Development: By Example***. Addison-Wesley Longman, 2002. 2. Kleinberg and Tardos – ***Algorithm Design***. Pearson Educational, 2014 (http://www.cs.princeton.edu/~wayne/kleinberg-tardos/) 3. Martin Fowler - ***Refactoring. Improving the Design of Existing Code***. Addison-Wesley, 1999. (http://refactoring.com/catalog/index.html) 4. ***The Python language reference***. (https://docs.python.org/3/reference/index.html) 5. ***The Python standard library***. (https://docs.python.org/3/library/index.html) 6. ***The Python tutorial***. (https://docs.python.org/3/tutorial/index.html)	
8.2 Seminar / laborator	Metode de predare
1. Prezentarea cursului și a limbajului Python	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică
2. Recursivitatea. Complexitatea algoritmilor	
3. Căutare. Sortare	
4. Metode de rezolvare a problemelor: greedy, divide et impera, backtracking, programarea dinamică	
5. Programarea procedurală	
6. Programarea modulară	
7. Clase și obiecte	
8. Programarea bazată pe obiecte	
9. Proiectarea programelor. Arhitectura stratificată	
10. Pregătirea examenului	
Bibliografie 1. Kent Beck - ***Test Driven Development: By Example***. Addison-Wesley Longman, 2002. 2. Kleinberg and Tardos – ***Algorithm Design***. Pearson Educational, 2014 (http://www.cs.princeton.edu/~wayne/kleinberg-tardos/) 3. Martin Fowler - ***Refactoring. Improving the Design of Existing Code***. Addison-Wesley, 1999. (http://refactoring.com/catalog/index.html) 4. ***The Python language reference***. (https://docs.python.org/3/reference/index.html) 5. ***The Python standard library***. (https://docs.python.org/3/library/index.html) 6. ***The Python tutorial***. (https://docs.python.org/3/tutorial/index.html)	

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

- Cursul respectă curriculumul IEEE și ACM pentru domeniul Informatică.
- Cursul există în programele de studiu ale universităților importante din România și din străinătate.
- Conținutul disciplinei este considerat de majoritatea companiilor software ca fiind deosebit de important pentru obținerea unor abilități avansate de programare.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Corectitudinea și volumul cunoștințelor acumulate, demonstrate prin implementarea corectă a programelor în limbajul Python.	Examen scris	**30%**
10.5 Seminar/laborator	Competențe legate de proiectarea, implementarea, testarea și depanarea programelor Python.	Evaluare practică	**30%**
	Corectitudinea rezolvării temelor de laborator și a documentației	Portofoliul de programe și documentație	**40%**

	aferente.		
10.6 Standard minim de performanță			
• Studenții vor respecta standardele de integritate academică. • Studenții trebuie să demonstreze o bună înțelegere a conceptelor prezentate și exersate în timpul cursului, că pot utiliza aceste cunoștințe într-un mod coerent, că pot forma anumite conexiuni și pot utiliza cunoștințele și competențele dobândite pentru rezolvarea mai multor tipuri de probleme de programare. • Pentru a intra în procesul de evaluare desfășurat pe durata sesiunii de examene sau de restanțe, studenții trebuie să cumuleze un număr de 10 prezențe la seminar (din 14 posibile) și 12 prezențe la laborator (din 14 posibile). • Pentru a intra în procesul de evaluare desfășurat pe durata sesiunii de examene, studenții trebuie să aibă nota minimă de 5 la laboratorul desfășurat pe durata semestrului. • Pentru a promova examenul, studenții trebuie să obțină o notă mai mare sau egală cu 5 în cadrul activităților de laborator, a testului practic și a examenului scris.			

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)[1]

Nu se aplică.

Data completării: 28.04.2025	Semnătura titularului de curs Conf. Univ. Dr. Molnar Arthur-Jozsef	Semnătura titularului de seminar Conf. Univ. Dr. Molnar Arthur-Jozsef

Data avizării în departament: ...	Semnătura directorului de departament Conf.dr. Adrian STERCA

[1] 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.*".