

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea de Vest din Timișoara
1.2 Facultatea / Departamentul	Facultatea de Chimie, Biologie, Geografie
1.3 Departamentul	Departamentul de Geografie
1.4 Domeniul de studii	Științe aplicate
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Geoinformatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare 1						
2.2 Titularul activităților de curs	Lect. Dr. Brîndușescu Alin						
2.3 Titularul activităților de seminar	Inf. Badiu Ioana						
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6 Tipul de evaluare	E ¹	2.7 Regimul disciplinei	DO

3. Timpul total estimat (ore pe semestru al activităților didactice)²

3.1 Număr de ore pe săptămână din care:	4	3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ din care:	56	3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate / pe teren					10
Pregătire seminare / laboratoare, teme, referate, portofolii și eseuri					30
Tutorat					10
Examinări					14
Alte activități					
3.7 Total ore studiu individual	94				
3.8 Total ore pe semestru	150				
3.9 Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	•
4.2 de competențe	• Cunoștințe elementare de logică și abilități de rezolvare a problemelor

5. Condiții (acolo unde este cazul)

¹ Conform articolului 37, alineatul (1) din Legea învățământului superior nr. 199/2023, cu modificările și completările ulterioare, „succesul academic al unui student pe parcursul unui program de studii este determinat prin **verificarea dobândirii rezultatelor așteptate ale învățării prin evaluări de tip examen și prin evaluarea pe parcurs**”.

5.1 de desfășurare a cursului	Față în față: Sală de curs cu tablă, videoproiector, conexiune la interne
5.2 de desfășurare a seminarului / laboratorului	Față în față: Sală de laborator dotată corespunzător, conexiune la internet

6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

Cunoștințe	C1. Cunoștințe fundamentale de informatică și matematică C2. Concepte referitoare la structura și funcționarea unui sistem de calcul C3. Concepte și metodologii privind analiza, proiectarea și implementarea aplicațiilor informatice
Abilități	A1. Abilitatea de a utiliza sisteme de fișiere, de a gestiona procesele specifice unui sistem de calcul, de a asigura comunicarea eficientă între componente software. A2. Abilitatea de a identifica algoritmi și structuri de date adecvate unei probleme concrete, de a aplica principiile de dezvoltare a unei aplicații informatice și de a implementa algoritmi într-un limbaj de programare. A3. Abilitatea de a utiliza medii/instrumente/platforme de programare specifice fiecărei etape din dezvoltarea unui sistem informatic.
Responsabilitate și autonomie	R1. Capacitatea de a rezolva în manieră autonomă sarcini specifice. R2. Capacitatea de a identifica/selecta soluții/căi de rezolvare adecvate și de a genera idei inovative. R3. Capacitatea de a identifica și planifica corect/eficient sarcinile specifice unui anumit proiect. R4. Capacitatea de a gestiona în manieră eficientă resursele implicate în realizarea unui proiect. R5. Capacitatea de a asuma în mod responsabil sarcinile profesionale și de a respecta normele de etică și deontologie profesională. R6. Capacitatea de a se adapta la noi cerințe și modalități de desfășurare a activității.

7. Conținuturi

Platforma prin care pot fi accesate suportul de curs în format electronic și alte resurse de învățare/bibliografice: google classroom

7.1 Curs	Metode de predare	Observații
Curs 1: Ce este informatica ? Concepte de baza privind sistemele de calcul. Etapele rezolvării unei probleme.Limbaje de programare.	Prelegere, conversație, exemplificare	

Curs 2: Operatori. Tipuri de baza în Python, expresii. Conversii de tip.	Prelegere, conversație, exemplificare	
Curs 3: Procesul de dezvoltare software. Cicluri: for, while, break, continue.	Prelegere, conversație, exemplificare	
Curs 4: Șiruri de caractere. Operații, conceptul de slicing. Expresii regulate.	Prelegere, conversație, exemplificare	
Curs 5-6: Structuri de date si secvențe: liste, tuple, dicționare, seturi.	Prelegere, conversație, exemplificare	
Curs 7: Funcții. Apelul funcțiilor, contextul de apel, variabile locale, variabile globale	Prelegere, conversație, exemplificare	
Curs 8: Module. Documentația programelor.	Prelegere, conversație, exemplificare	
Curs 9: I/O In Python. Fluxuri, Fișiere. Structurarea informațiilor în TXT, CSV.	Prelegere, conversație, exemplificare	
Curs 10: I/O In Python. Fluxuri, Fișiere. Structurarea informațiilor în XML, JSON, YAML.	Prelegere, conversație, exemplificare	
Curs 11-13: Tipuri de date abstracte: Clase, Obiecte. Interacțiunea dintre obiecte. Mostenire, Incapsulare, etc. Reutilizare cod.	Prelegere, conversație, exemplificare	
Curs 14: Legarea tuturor conceptelor.	Prelegere, conversație, exemplificare	
Bibliografie: [1] Z. Shaw, <i>Learn Python : The Hard Way (3rd Edition)</i> , 2013. [2] D. Beazley, Brian K. Jones, <i>Python Cookbook, Third edition</i> , O'Reilly Media, 2013. [3] M. Lutz, <i>Learning Python</i> , 2007. doi:10.1016/0019-1035(89)90077-8. [4] M. Summerfield, <i>Programming in Python 3</i> , 2010. doi:9788441526136. [5] J.M. Zelle, <i>Python programming: an introduction to computer science</i> , Franklin, Beedle & Associates, Inc., 2004. [6] D. Bader, <i>Python Tricks: A Buffet of Awesome Python Features</i> , Dan Bader, 2017.		
7.2 Seminar / laborator	Metode de predare	Observații
Laborator 1: Exerciții de gândire algoritmică.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	

Laborator 2: Expresii simple, Input, output, introducerea de date simple.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 3: Aplicații geografice simple.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 4: Introducere concupse șiruri de caractere. Exerciții simple.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 5: Introducere concept list comprihansion. Exerciții simple cu liste.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 6: Exerciții simple cu dicționare și seturi.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 7: Exerciții simple cu funcții.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 8: Test de laborator	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 9: Exerciții cu seturi de date.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 10: Construirea de fișiere JSON clasice și introducere în concept de GeoJSON.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 11-13: Exerciții simple OOP.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Laborator 14: Test de laborator.	Problematizare, dialog, învățare prin colaborare, învățare prin descoperire.	
Bibliografie:		
Idem cu cea de la curs		

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

Conținutul este în concordanță cu structura cursurilor similare de la alte universități și acoperă aspectele fundamentale necesare familiarizării cu problematica programării orientate obiect. Abilitatea de a identifica, proiecta, implementa și analiza probleme care se pot rezolva folosind principii orientate obiect este esențială pentru orice activitate din domeniul informaticii. Competențele oferite de această disciplină sunt necesare unui specialist IT pentru a identifica soluții de rezolvare a unor probleme concrete, indiferent de domeniul specific de activitate.

9. Utilizarea instrumentelor bazate pe inteligența artificială generativă

Studentii pot utiliza instrumente de inteligență artificială generativă doar ca suport teoretic (explicații, exemple simple de sintaxă). Generarea de cod sursă pentru teme, laboratoare sau evaluări nu este permisă. Orice cod prezentat trebuie să fie rezultatul muncii individuale a studentului. Copierea codului de la AI poate bloca formarea competențelor fundamentale.

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 coerența răspunsurilor la întrebările teoretice; - Înțelegerea conceptelor fundamentale de programare procedurală și orientată obiect; - Capacitatea de a analiza fragmente de cod și de a explica comportamentul acestora; - Rezolvarea corectă a problemelor de programare, respectând cerințele	Examen scris în sesiunea de examene - obligatoriu	30 %
10.5 Laborator	Teste de laborator in timpul semestrului	Evaluare practică	70%
10.6 Standard minim de performanță			
Nota finală se calculează ca medie ponderată a notelor acordate pentru componentele specificate la 10.4 și 10.5. Examenul se consideră promovat dacă media este cel puțin 5.00. Dacă media testelor de laborator nu este peste 5 la finalul semestrului atunci va trebui recontractată materia. La fiecare dintre sesiunile de examen (inclusiv cele de restanță și măriri) nota se calculează după aceeași regulă. În sesiunea de restanțe/măriri se poate reda doar examenul scris din sesiune. Regulile referitoare la prezență sunt cele din regulamentele UVT privind frecventarea activității de laborator.			

Data completării

29.09.2026

Titular de disciplină:

Lect. Dr. Brîndușescu Alin

Semnătura:

Data avizării în departamente

Director de departament