

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Tibiscus” din Timișoara
1.2. Facultatea	Calculatoare și Informatică Aplicată
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii/Calificarea	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	PROGRAMARE II (P2) – LIN121						
2.2. Titularul activității de curs	Conf.univ.dr. Florentina Anica PINTEA						
2.3. Titularul activității de laborator	Dan PĂTRAȘCU-BABA						
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Conținut=DF Obligatorietate=DO

3. Timpul total estimat

3.1. Numărul de ore pe săptămână	4	din care 3.2. curs	2	3.3. laborator	2
3.4. Total ore din planul de învățământ	56	din care 3.5. curs	28	3.6. laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren					23
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					-
Examinări					4
Alte activități					-
3.7. Total ore studiu individual					69
3.8. Total ore pe semestru					125
3.9. Numărul de credite					5

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Programare 1, Fundamente de informatică aplicată Cunoștințe de bază privind conceptele fundamentale ale programării: variabile, tipuri de date, structuri de control, funcții.
4.2. de competențe	Capacitatea de a elabora algoritmi simpli. Abilități de rezolvare a problemelor utilizând un limbaj de programare. Capacitatea de a elabora, testa și depana programe simple.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală dotată cu videoproiector, Calculator și software specific
5.2. de desfășurare a laboratorului	Laborator informatic, Python instalat Mediu de dezvoltare pentru limbajul Python (VS Code sau similar)

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

Cunoștințe	La finalul disciplinei, studentul va fi capabil: - să descrie conceptele fundamentale ale programării avansate în limbajul Python; - să explice utilizarea structurilor de date (liste, tuple, seturi, dicționare); - să explice modul de definire și utilizare a funcțiilor; să descrie modalitățile de utilizare a fișierelor în Python.
------------	--

Abilități	La finalul disciplinei, studentul va fi capabil: - să elaboreze algoritmi pentru rezolvarea problemelor; - să implementeze programe utilizând limbajul Python; - să identifice și să corecteze erorile din programe; - să testeze și să valideze aplicațiile realizate; - să dezvolte aplicații de complexitate medie.
Responsabilitate și autonomie	La finalul disciplinei, studentul va fi capabil: - să utilizeze eficient resursele disponibile pentru rezolvarea problemelor; - să dezvolte aplicații în mod independent; - să respecte cerințele specificate; - să lucreze individual și în echipă; - să își autoevalueze rezultatele și să își îmbunătățească performanțele.

7. Conținuturi

7.1. Curs	Metode de predare/ învățare	Observații (28 ore)
1. Introducere în limbajul Python. Caracteristici. Domenii de utilizare	expunere interactivă, problematizare, discuții	Prezentarea rolului limbajului Python în dezvoltarea aplicațiilor moderne și a domeniilor de utilizare
2. Medii de dezvoltare Python. VS Code	demonstrație, studiu de caz	Familiarizarea studenților cu mediile de dezvoltare utilizate în practică
3. Elemente fundamentale ale limbajului Python	expunere, exemple, aplicații	Formarea competențelor de bază pentru scrierea programelor
4. Excepții. Tratarea excepțiilor	explicație, exemple practice	Dezvoltarea capacității de gestionare a erorilor
5. Funcții. Funcții predefinite. Module și pachete	expunere, aplicații	Dezvoltarea programelor modulare
6. Core Built-ins și module standard de bibliotecă	studiu de caz, aplicații	Utilizarea eficientă a funcțiilor și modulelor standard
7. Șiruri de caractere. Funcții și metode asociate	demonstrație, exerciții	Aplicarea tehnicilor de procesare text
8. Operații cu fișiere și text	expunere, exemple, aplicații	Dezvoltarea aplicațiilor cu stocare persistentă
9. Operații cu Time	expunere, exemple, aplicații	Utilizarea datelor temporale în aplicații
10. Procesare numerică. Introducere în NumPy	expunere, exemple, aplicații	Introducere în procesarea numerică și analiza datelor
11. Bazele comunicației în rețea. Socket programming. Principii de rețea. Interfața Berkeley Socket	expunere, demonstrație	Înțelegerea principiilor comunicației în rețea
12. Module de protocol de rețea pe partea client. Protocoale de email	expunere interactivă, problematizare, discuții	Dezvoltarea aplicațiilor client
13. Text structurat: HTML	demonstrație, exerciții	Introducere în procesarea documentelor Web
14. Grafica în Python. Utilizarea modulelor matplotlib și graphics	expunere, exemple, aplicații	Vizualizarea datelor și dezvoltarea interfețelor grafice
7.2. Laborator/Proiect (teme de laborator, proiecte etc.)	Metode de predare- învățare	Observații (28 ore)
1. Introducere în limbajul Python. Instalare și configurare	explicație, exerciții practice	Configurarea mediului de lucru
2. Medii de dezvoltare Python	aplicații practice	Utilizarea mediului de dezvoltare
3. Limbajul Python	exerciții, implementare	Formarea deprinderilor de programare
4. Excepții	exerciții practice	Tratarea erorilor

5. Funcții. Funcții predefinite. Module.	implementare programe	Dezvoltarea aplicațiilor modulare
6. Core Built-ins și module standard de bibliotecă	aplicații practice	Utilizarea funcțiilor standard
7. Șiruri de caractere. Funcții și metode asociate	exerciții	Procesarea textului
8. Operații cu fișiere și text	implementare programe	Aplicații cu fișiere
9. Operații cu Date/ Time	aplicații practice	Utilizarea timpului
10. Procesare numerică	aplicații practice	Aplicații numerice
11. Bazele rețelei. Principii de rețea. Interfața Berkeley Socket	aplicații client	Dezvoltarea aplicațiilor de rețea
12. Module de protocol de rețea pe partea client. Protocole de email	studiu de caz	Aplicații client
13. Text structurat: Procesare HTML	aplicații practice	Aplicații Web
14. Grafica în Python. Utilizarea modulelor matplotlib și graphics	Prezentare, discuții	Prezentarea proiectului și evaluare

Bibliografie

- [1] M. Lutz, *Learning Python*, O'Reilly (ediția a 5-a), 2025 / <https://learning-python.com/about-lp6e>
- [2] Alex Martelli, Anna Ravenscroft, and Steve Holden, *Python in a Nutshell, Third Edition*, O'Reilly, 2017
- [3] B. Miller, D. Ranum, *Problem Solving with Algorithms and Data Structures using Python*, 2023
- [4] Eric Matthes, *Python Crash Course*, Beginners to intermediate Python learners, No Starch Press, 2023
- [5] M.T. Goodrich, R. Tamassia, M.H. Goldwasser, *Data Structures and Algorithms in Python*, Wiley, 2013
- [6] M. Summerfield, *Programming in Python 3*, Addison-Wesley, 2009
- [7] J. Kleinberg, E. Tardos, *Algorithm Design*, Addison-Wesley, 2005
- [8] The Python language reference. <http://docs.python.org/py3k/reference/index.html>
- [9] The Python standard library. <http://docs.python.org/py3k/library/index.html>
- [10] The Python tutorial. <http://docs.python.org/tutorial/index.html>

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

Conținutul disciplinei este în concordanță cu cerințele actuale ale pieței muncii din domeniul informaticii, precum și cu recomandările comunității academice și ale organizațiilor profesionale din domeniu. Se asigură cunoștințe teoretice și practice pentru realizarea de aplicații practice utilizând algoritmi și structuri de date în Python. Limbajul Python, utilizat în cadrul disciplinei, este unul dintre cele mai răspândite limbaje de programare la nivel internațional, fiind folosit în dezvoltarea aplicațiilor software, aplicațiilor web, analiza datelor, inteligență artificială și automatizarea proceselor. Prin parcurgerea acestei discipline, studenții dobândesc cunoștințe și abilități necesare pentru dezvoltarea aplicațiilor software de complexitate medie și pentru continuarea formării în domenii avansate.

Disciplina contribuie, totodată, la dezvoltarea capacității studenților de a utiliza eficient resursele disponibile, de a lucra independent și de a se adapta la cerințele în continuă schimbare ale domeniului IT.

9. Evaluare

Tip de activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
9.1. Curs	- cunoașterea conceptelor teoretice prezentate la curs; - înțelegerea principiilor programării în Python; - capacitatea de analiză și rezolvare a problemelor specifice; - participarea activă la activitățile didactice	Examen scris (20%) + Prezentarea unei teme de cercetare din domeniul disciplinei (30%)	50%
9.2. Laborator	Activitate laborator	Evaluarea activităților de laborator	10%
	- capacitatea de aplicare practică a cunoștințelor; - rezolvarea corectă a aplicațiilor propuse; - implicarea și participarea activă la laborator; - calitatea temelor și a proiectului realizat	Evaluarea rezolvării temelor de laborator Evaluare practică (test practic și/sau proiect)	40%

9.3. Standard minim de performanță

Pentru obținerea notei 5–6, studentul trebuie să demonstreze cunoașterea noțiunilor fundamentale ale limbajului Python și să înțeleagă conceptele de bază privind structurile de date, funcțiile și lucrul cu fișiere. Studentul este capabil să realizeze programe simple, cu un nivel minim de complexitate, și să rezolve parțial cerințele propuse, cu sprijin.

Pentru obținerea notei 7–8, studentul demonstrează o bună cunoaștere a conceptelor studiate și capacitatea de a le explica și utiliza în mod corect. Acesta poate realiza independent programe de complexitate medie, utilizând adecvat structurile de date, funcțiile și modulele studiate. Studentul rezolvă corect majoritatea problemelor propuse.

Pentru obținerea notei 9–10, studentul demonstrează o cunoaștere foarte bună și aprofundată a conceptelor studiate și capacitatea de a le aplica în mod corect, eficient și creativ. Acesta realizează programe complete, corecte și bine structurate, rezolvă integral cerințele propuse și demonstrează capacitate de analiză, sinteză și lucru independent. Studentul manifestă inițiativă, implicare activă și capacitatea de a dezvolta aplicații de complexitate ridicată, utilizând în mod adecvat conceptele și tehnologiile studiate.

Data completării

Titular de curs

Titular de laborator

16.02.2026

Conf.univ.dr. Florentina -Anica PINTEA

Dan PĂTRAȘCU-BABA

Data avizării în departament

Director de departament

20.02.2026

Conf.univ.dr. Victoria IORDAN