



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie POLITEHNICA București
1.2 Facultatea	Științe Aplicate
1.3 Departamentul	Matematică-Informatică
1.4 Domeniul de studii universitare	Științe Inginerești Aplicate
1.5 Programul de studii universitare	Matematică și Informatică Aplicată în Inginerie
1.6 Ciclu de studii universitare	Licență
1.7 Limba de predare	Română
1.8 Locația geografică de desfășurare a studiilor	București

2. Date despre disciplină

2.1 Denumirea disciplinei							
(ro)	Python cu aplicații în inginerie						
(en)	Python with applications in engineering						
2.2 Titularul activităților de curs	Conf. Univ. Dr. MARICA Aurora-Mihaela						
2.3 Titularul activităților de laborator/seminar	Drd. STANCIU Monica, IRIMIA Ani						
2.4 Anul de studiu	2	2.5 Semestrul	II	2.6. Tipul de evaluare	V	2.7 Statutul disciplinei	Ob
2.8 Categoria formativă	DD	2.9 Codul disciplinei	UPB.13.D.04.O.137				

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

3.1 Număr 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:					
Studiul după manual, suport de curs, bibliografie și notițe					6
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					6
Pregătire teme și referat					28
Tutorat					2
Examinări					2
3.7 Total ore studiu individual	44				
3.8 Total ore pe semestru	100				
3.9 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Parcursarea și promovarea disciplinei Programarea calculatoarelor și limbaje de programare 1/2 și POO, în care se însușesc noțiuni și se formează deprinderi de utilizare a limbajelor de programare C++ și Java.
4.2 de rezultate ale învățării	Acumularea unor noțiuni generale de programare cum ar fi instrucțiune sau structuri de control (if, for, while)

5. Condiții necesare pentru desfășurarea optimă a activităților didactice

5.1 Curs	Cursul se va desfășura într-o sală dotată cu videoproiector și conexiune la internet pentru calculatorul/laptopul cadrului didactic
5.2 Laborator	Laboratoarele se vor desfășura într-o sală cu dotare specifică, care trebuie să includă programele Python și un editor de cod/IDE (VS Code sau PyCharm) instalate pe calculatoarele studenților și conexiune la internet

6. Obiectiv general. Această disciplină se studiază în cadrul specializării Matematică și Informatică Aplicată în Inginerie și își propune să familiarizeze studenții cu principalele instrumente/concepte necesare programării în Python: tipuri de date, colecții (liste, dicționare), instrucțiuni condiționale if și iterative (for, while), introducerea datelor de la tastatură, scrierea funcțiilor și organizarea codului în clase pentru a face părți din cod reutilizabil, tratarea claselor principale de erori din cod. De asemenea, studenții vor învăța și cum să realizeze diverse proiecte pornind de la noțiunile generale, cum ar fi dezvoltarea unor jocuri, tehnici de vizualizare a datelor, aplicații web.

7. Rezultatele învățării

Cunoștințe	C1. Identifică și descrie concepte, principii și metode de bază din informatică. C2. Explică și interpretează rezultate teoretice și experimentale din informatică. C8. Recunoaște, explică și specifică fundamentele matematice aplicate în informatică. C9. Alege, descrie, analizează, explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă. • C1: Să creeze/printeze stringuri formate care folosesc variabile în interiorul stringurilor. • C1: Să adauge spații în stringuri cu escape sequences: tab \t și newline \n. • C1: Să creeze/folosească variabile numerice întregi sau cu virgulă mobilă. • C1: Să creeze liste, să apeleze/modifice/șteargă/sorteze/extragă componente din ele. • C1: Se creeze liste imutabile/tuples. • C1: Să folosească corect instrucțiuni if-elif-else, while. • C1: Să definească perechi cheie-valoare/dicționare, să le modifice, șteargă, acceseze, parcurgă. • C1: Să scrie instrucțiuni care cer date de la tastatură. • C1: Să creeze funcții/clase. • C1: Să citească date dintr-un fișier extern. • C1: Să folosească corect instrucțiuni try-except pentru a trata clase principale de excepții. • C2: Să înțeleagă instrucțiunile unui joc creat în Python de alți programatori. • C2: Să creeze jocuri care să respecte anumite cerințe asupra butoanelor. • C2, C8: Să ploteze/deseneze graficul unei funcții, respectând cerințele asupra culorii, titlului, label-urilor, markere, legendă, folosind pachetele turtle și matplotlib. • C2: Să creeze o aplicație web simplă folosind web framework-ul Django. • C8: Să lucreze cu array-uri folosind pachetul numpy.
Abilități	• A1. Operează cu concepte, principii și metode de bază din informatică. • A7. Utilizează metode și instrumente specifice pentru studiul, analiza, sinteza și crearea aplicațiilor în Python. • A8. Diagnostichează/depanează blocuri de cod de complexitate mică/medie în Python. • A9. Creează aplicații funcționale de complexitate mică/medie în Python.

Responsabilitate și autonomie

- R1. Aplică valorile eticii și deontologiei profesiei de inginer: Respectă principiile de etică academică, citând corect sursele bibliografice utilizate.
- R4. Practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor.
- R5. Comunică eficient despre activitățile de programare cu o gamă largă de public.
- R6. Este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate. Identifică oportunități de formare continuă și de utilizare eficientă, pentru propria dezvoltare, a surselor informaționale (portaluri Internet, aplicații software de specialitate, baze de date, cursuri on-line etc.), atât în limba română, cât și într-o limbă de circulație internațională.
- R7. Selectează/analizează surse bibliografice potrivite proiectelor pe care le realizează.
- R8. Produce software și îl adaptează continuu la noile tehnologii și cerințe de piață. Analizează/valorifică oportunități de afaceri în domeniul programării.
- R9. Demonstrează autonomie în învățare și în organizarea situației/contextului de învățare sau a situației-problemă de rezolvat.
- R11. Își asumă în mod responsabil sarcinile profesionale și respectă normele de etică și deontologie profesională: Demonstrează receptivitate pentru contexte noi de învățare.
- R12. Lucrează eficient ca membru în echipă sau lider al acesteia. Manifestă colaborare cu ceilalți colegi/profesorii în desfășurarea activităților didactice. Demonstrează abilități de management al situațiilor din viața reală (gestionarea timpului, colaborare/conflict)

8. Metode de predare

Pornindu-se de analiza caracteristicilor de învățare ale studenților și de la nevoile lor specifice, procesul de predare va explora metode de predare atât expositive (prelegerea, expunerea, prezentarea), cât și conversative-interactive (conversația euristică, problematizarea, modelarea), bazate pe modele de învățare prin descoperire facilitate de explorarea directă și indirectă a realității (exemplificarea, demonstrația), dar și pe metode bazate pe acțiune, precum activitățile practice.

În activitatea de predare vor fi utilizate prelegeri la tablă, asociate când este cazul cu prezentări Power Point sau diferite filmulețe care vor fi puse la dispoziția studenților. Fiecare curs va debuta cu recapitularea capitolelor deja parcurse, cu accent asupra noțiunilor parcurse la ultimul curs. Prezentările utilizează și imagini/scheme, astfel încât informațiile prezentate să fie mai ușor de înțeles/asimilat.

Această disciplină acoperă informații și activități practice menite să-i sprijine pe studenți în dezvoltarea unor relații de colaborare și comunicare într-un climat favorabil învățării prin descoperire.

Se va avea în vedere exersarea abilităților de ascultare activă și de comunicare asertivă, precum și a mecanismelor de construcție a feedback-ului, ca modalități de reglare comportamentală în situații diverse și de adaptare a demersului pedagogic la nevoile de învățare ale studenților.

Se va exersa abilitatea de lucru în echipă pentru rezolvarea diferitelor sarcini de învățare.

9. Conținuturi

CURS		
CAPITOL	CONȚINUT	#ORE
I.	Caracteristici ale limbajului Python. Explicarea primului program	2
II.	Tipuri numerice de date	2
III.	Utilizarea/generarea graficelor în programe	2
IV.	Stringuri, liste	2
V.	Importarea și prelucrarea datelor din fișiere	2
VI.	Definirea funcțiilor	2
VII.	Funcții și interfețe	2
VIII.	Structuri/instrucțiuni decizionale	2
IX.	Date booleene. Structuri/instrucțiuni iterative/de tip loop	2
X.	Abordarea probabilităților în Python	2
XI.	Clase și obiecte	2

XII.	Colecții	2
XIII.	Structuri/instrucțiuni decizionale	2
XIV.	Moștenire	2

Bibliografie

1. A.B. Downey, Think Python, O'Reilly, 2024
2. M. Lutz, Learning Python, O'Reilly, 2018.
3. P. Barry, Head First Python, O'Reilly, 2023.
4. E. Matthes, Python Crash Course, Nostarch Press, 2023.
5. J. Zelle, Python Programming, Franklin Beedle, 2017.
6. C. Hill, Learning Scientific Programming with Python, Cambridge University Press, 2020.
7. A. Yu, Master Python by Building 100 projects in 100 days, curs Udemy.
8. M. Hamedani, Complete Python Mastery, curs online <https://codewithmash.com/courses>

LABORATOR

CAPITOL	CONȚINUT	#ORE
I.	Tipuri de date simple, variabile și liste	2
II.	Instrucțiuni if și dicționare	2
III.	User input, instrucțiuni iterative while	2
IV.	Funcții și clase	2
V.	Excepții	2
VI.	Parcurgerea jocului Aliens	2
VII.	Crearea unui joc cu cerințe	2
VIII.	Grafice de funcții	2
IX.	Downloadarea datelor	2
X.	Folosirea API-urilor Git și GitHub	2
XI.	Crearea unei aplicații web cu frameworkul Django	2
XII.	Crearea unei aplicații web cu frameworkul Django	2
XIII.	Crearea unei aplicații web cu frameworkul Django	2
XIV.	Testare laborator	2

Bibliografie

1. A. Sweigart, Invent your own computer games with Python, Nostarch Press, 2017.
2. D. Kopec, Classic Computer Science Problems in Python, Manning, 2019.
3. L. Vaughan, Python Tools for Scientists, Nostarch Press, 2023.
4. A. Yu, Master Python by Building 100 projects in 100 days, curs Udemy.

9. Evaluare. Activitatea fiecărui student va fi evaluată cu maxim 100p, din care:

- 10p prezența curs+laborator
- Parcurgerea proiectelor de laborator și efectuarea temelor 60p
- Testare în timpul ultimului laborator 20p
- Răspunsuri/activitate curs 10p

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere în nota finală
10.1.Curs	SC=realizare sarcini curs	Conversația euristică	Total curs=15 (SC=10, prezența=5)
10.2L.Laborator	SL=realizare sarcini laborator	Conversația euristică	Total laborator=85 (SL=60, prezența=5, testare=20)
		Depozitare Moodle	
		Depozitare Moodle	
10.3 Standard minim de performanță			
• Pentru promovarea disciplinei, studentul trebuie să obțină cel puțin 50 de puncte din 100. • Participarea la toate laboratoarele (recuperarea zilelor de absență). Participarea la testarea din cadrul ultimului laborator.			



Universitatea Națională de Știință și Tehnologie

POLITEHNICA București

Facultatea de Științe Aplicate



Data completării:
22.09.2025

Semnătura titularului de curs:
Conf. Univ. dr. MARICA Aurora-Mihaela

Semnătura titularilor de laborator:
Drd. STANCIU Monica, IRIMIA Ani

Data aprobării în
Consiliul Facultății

Semnătura decanului Facultății de Științe Aplicate:
Conf.dr. PETRESCU-NIȚĂ Alina