

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea ¹ / Departamentul ²	Electronică, Telecomunicații și Tehnologii Informaționale / Comunicații
1.3 Domeniul de studii (denumire/cod ³)	Inginerie electronică, telecomunicații și tehnologii informaționale (20/20/10/100)
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Tehnologii și Sisteme de Telecomunicații / 20/20/10/100/20 / Tehnologii și Sisteme de Telecomunicații

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă ⁴	Limbaje de programare 3 / DF						
2.1b Denumirea disciplinei în limba engleză	Programming Languages 3						
2.2 Titularul activităților de curs	Sl.dr.ing. Drăgulescu Bogdan, Conf.dr.ing. Mocofan Muguraș						
2.3 Titularul activităților aplicative ⁵	Sl.dr.ing. Drăgulescu Bogdan, Conf.dr.ing. Mocofan Muguraș, As.dr.ing. Mihăilescu Dan Mihai						
2.4 Anul de studii ⁶	2	2.5 Semestrul	4	2.6 Tipul de evaluare	V	2.7 Regimul disciplinei ⁷	DOB

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁸

3.1 Număr de ore asistate integral/săptămână	4 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	0/2/0
3.1* Număr total de ore asistate integral/sem.	56 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	0/28/0
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	3.14 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1.14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1
3.7* Număr total de ore activități neasistate/semestru	44 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			16
		ore studiu individual după manual, suport de curs, bibliografie și notițe			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			14
3.8 Total ore/săptămână ⁹	7.1				
3.8* Total ore/semestru	100				
3.9 Număr de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• -
4.2 de rezultatele învățării	• Competențe de bază în utilizarea și programarea calculatoarelor.

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	• Sală cu proiector și tablă.
5.2 de desfășurare a activităților practice	• Laborator de informatică cu suficiente calculatoare pentru numărul de studenți.

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	<ul style="list-style-type: none"> • C1. Studentul/absolventul identifică și descrie concepte, principii și metode de bază din matematică, fizică, grafică asistată de calculator, bazele electrotehnicii, limbaje de programare. • C2. Studentul/absolventul explică și interpretează rezultate teoretice și experimentale din matematică, fizică, analiza și sinteza circuitelor, programarea calculatoarelor, și grafica asistată de calculator. • C5. Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la arhitectura sistemelor de calcul, microcontrolere, limbaje și tehnici de programare și modul lor de aplicare în probleme concrete.
Abilități	<ul style="list-style-type: none"> • A1. Studentul/absolventul operează cu concepte, principii și metode de bază din matematică, fizică, grafică asistată de calculator, bazele electrotehnicii, limbaje de programare. • A17. Studentul/absolventul specifică cerințe, elaborează programe în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), execută, depanează și interpretează rezultatele programelor realizate în vederea rezolvării unei probleme concrete. • A19. Studentul/absolventul elaborează și rezolvă exerciții practice, lucrări de laborator și probleme aplicative, demonstrând capacitatea de integrare a noțiunilor teoretice.
Responsabilitate și autonomie	<ul style="list-style-type: none"> • RA2. Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor. • RA3. Studentul/absolventul comunică eficient despre activitățile de inginerie cu o gamă largă de public. • RA6. Studentul/absolventul lucrează eficient ca membru în echipă sau lider al acesteia. • RA10. Studentul/absolventul manifestă capacitatea de autoorganizare și de gestionare a timpului de studiu, respectând cerințele și termenele activităților academice.

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

<ul style="list-style-type: none"> • Disciplina are ca scop consolidarea și extinderea cunoștințelor studenților privind conceptele fundamentale ale limbajelor de programare, esențiale în dezvoltarea de aplicații software pentru sisteme embedded, aplicații în rețea și prelucrarea datelor. Prin parcurgerea acestei discipline, studenții vor înțelege și aplica tipuri de date, structuri de control, funcții, structuri de date complexe, precum și paradigme moderne de programare (orientată pe obiect și funcțională). • Studenții vor dobândi competențe esențiale în scrierea, testarea și depanarea codului, în programarea concurentă și în utilizarea unor instrumente de testare automată. Totodată, disciplina vizează dezvoltarea capacității studenților de a elabora programe în limbaje de uz general, de a organiza codul sursă în module și pachete, de a lucra cu fișiere și de a implementa aplicații care interacționează cu rețele de comunicații sau servicii web.
--

8. Conținuturi¹⁰

8.1 Curs	Număr de ore	Metode de predare ¹¹
Limbaje de programare. Introducere.	2	Prelegere participativa, problematizare, dezbatere, verificare. Utilizarea resurselor educaționale deschise, consultații prin intermediul platformelor electronice (email, Microsoft Teams).
Noțiuni de bază ale limbajului de programare. Tipuri de date. Variabile și expresii. Structuri de control. Funcții.	2	
Structuri de date de tip secvență – liste, tupluri, șiruri de caractere. Extragerea datelor pe baza expresiilor regulate.	3	
Structuri de date de tip mulțime. Structuri de date de tip dicționar.	3	
Paradigme de programare - OOP. Principii OOP. Clase. Instanțe. Moștenire. Polimorfism. Încapsulare. Diferențieri în implementare.	3	
Paradigme de programare - Elemente de programare funcțională. Expresii lambda. Map. Filter. Reduce. Iterator versus recursivitate.	3	
Operații cu fișiere. Serializarea datelor. Erori și excepții. Interceptarea și tratarea excepțiilor.	3	
Testarea automată a codului - Testarea unitară. Organizarea testelor. Module specializate pentru testare. Organizarea aplicațiilor folosind module și pachete.	3	
Programarea în rețea. Socket-uri. Accesarea datelor de pe Web. Clienți web. Servicii web și API-uri.	3	
Programare concurentă. Fire de execuție. Sincronizarea firelor de execuție. Multiprocesor. Sincronizarea proceselor	3	

Bibliografie¹² 1. M. Bucos, B. Dragulescu, Programare in Python, Editura Politehnica Timișoara, ISBN: 978-606-35-0607-9, 2025, disponibilă biblioteca UPT 2. C. Orhei, M. Mocoan, S. Vert, Limbaje de programare: aplicații practice folosind Python, Editura Politehnica Timișoara, ISBN: 978-606-35-0595-9, disponibilă biblioteca UPT 3. B. Drăgulescu, Resurse curs în format deschis, https://github.com/DataLabUPT/ccCourse 4. C. Severance, Python for Everybody: Exploring Data in Python, CreateSpace Independent Publishing Platform, 978-1-5300-5112-0, 2016, Open Book 5. A. Downey, Think Python, 3rd Edition. Beijing: O'Reilly Media, 2024, Open Book 6. ***, Python Documentation, https://docs.python.org/3/ , 2025		
8.2 Activități aplicative¹³	Număr de ore	Metode de predare
Introducere. Instalare. Lucrul cu interpretorul. Medii integrate de dezvoltare. Medii de dezvoltare izolate.	4	Expunere, discuție liberă, problematizare, aplicație practică, verificare. Utilizarea resurselor educaționale deschise.
First class citizens. Structuri de control. Funcții.	2	
Secvențe. Liste. Tupluri. Introducerea tipurilor de date de tip secvență. Operații specifice. Alegerea structurii optime pentru rezolvarea unei probleme.	2	
Mulțimi. Dicționare. Structuri de date de tip mulțime. Structuri de date de tip dicționar. Citirea de la tastatură.	2	
Procesarea șirurilor de caractere. Șiruri de caractere. Operații cu șiruri de caractere. Citirea unui fișier text. Utilizarea expresiilor regulate.	2	Expunere, discuție liberă, problematizare, verificare..
Programare funcțională. Funcții lambda. Utilizare Map, Filter, Reduce. Modulele destinate programării funcționale.	2	Expunere, discuție liberă, problematizare, verificare.
Fișiere. Citirea și scrierea în fișiere. Tratarea excepțiilor. Serializarea datelor. Formatele: JSON, CSV, Pickle.	2	Expunere, discuție liberă, problematizare, verificare.
Dezvoltarea de aplicații utilizând module specializate. Vizualizarea datelor. Fire de execuție. Programare paralelă.	8	Expunere, discuție liberă, problematizare, aplicație practică, verificare.
Evaluarea cunoștințelor dobândite prin rezolvarea unor probleme practice.	4	Aplicație practică, verificare
Bibliografie¹⁴ 1. B. Drăgulescu, Limbaje de programare 3 – activități practice, https://cv.upt.ro/ , 2025 2. C. Orhei, M. Mocoan, S. Vert, Limbaje de programare: aplicații practice folosind Python, Editura Politehnica Timișoara, ISBN: 978-606-35-0595-9, disponibilă biblioteca UPT 3. C. Severance, Python for Everybody: Exploring Data in Python, CreateSpace Independent Publishing Platform, 978-1-5300-5112-0, 2016, Open Book 4. A. Downey, Think Python, 3rd Edition. Beijing: O'Reilly Media, 2024, Open Book 5. ***, Python Documentation, https://docs.python.org/3/ , 2025		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ¹⁵	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Cunoașterea noțiunilor și conceptelor fundamentale prezentate la curs și laborator	Evaluare cunoștințelor se realizează prin examen scris distribuit în două părți. Partea teoretică este evaluată prin teste cu itemi de mai multe tipuri, iar partea practică prin itemi de tip grilă în care se dorește evaluarea unei secvențe scurte de cod. Implementarea evaluării se realizează în format electronic prin intermediul platformei Campus Virtual.	50%
9.5 Activități aplicative	S:		
	L: Aplicarea cunoștințelor pentru rezolvarea unor probleme. Rezolvarea cerințelor.	Evaluare cu ajutorul calculatorului prin rezolvarea unor probleme. Vor fi cel puțin două teste de acest tip. Suplimentar se realizează evaluare scurtă la finalul unor	50%

		laboratoare prin teste grilă implementate în platforma Campus Virtual.	
	P¹⁶:		
	Pr:		
9.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁷)			
<ul style="list-style-type: none"> • Scriere cod fără erori de sintaxă. Se verifică în cadrul testelor practice. • Alegerea corectă a structurilor de date pentru probleme specifice. Se verifică prin examen și teste practice. • Implementarea unor secvențe de cod de complexitate scăzută. Se verifică în cadrul testelor practice • Recunoașterea părților componente și rolul lor în cadrul unei secvențe de cod de complexitate medie. Se verifică prin examen. 			

Data completării

03.09.2025

**Titular de curs
(semnătura)**

**Titular activități aplicative
(semnătura)**

**Director de departament
(semnătura)**

Data avizării în Consiliul Facultății¹⁸

**Decan
(semnătura)**

07.10.2025