FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Inginerie Electrică și Știința Calculatoarelor
Departamentul	Calculatoare, electronică și automatică
Domeniul de studii	Automatică, informatică aplicată și sisteme inteligente
Ciclul de studii	Licență, dual
Programul de studii	Automatică și informatică aplicată

2. Date despre disciplină

Denumirea disciplinei	**Complemente de programare**				
Anul de studiu	**II**	Semestrul	**4**	Tipul de evaluare	**V**
Regimul disciplinei	Categoria formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				**DC**
	Categoria de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				**DFA**

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	Total general	2	Curs		Seminar	2	Laborator IIS		Proiect IIS		Practică IIS	
							Laborator IM		Proiect IM		Practică IM	
I b) Totalul de ore pe semestru din planul de învățământ		28	Curs		Seminar	28	Laborator		Proiect		Practică	

(*IIS – instituție de învățământ superior; IM – învățare prin muncă*)

Distribuția fondului de timp pe semestru:	Ore IIS	Ore IM
II. Studiul după manual, suport de curs, bibliografie și notițe	19	
III. Examinări	3	
IV. Alte activități (precizați):		

Total ore studiu individual (II+III)	Ore IIS	22	Ore IM	
Total ore pe semestru (I.b+II+III+IV)	Ore IIS	50	Ore IM	
Numărul de credite	Credite IIS	2	Credite IM	

4. Competențe specifice acumulate

Competențe transversale	CT5. aplică cunoștințe științifice, tehnologice și inginerești
Competențe profesionale	CP3. definește arhitectura software CP4. creează softuri CP5. dezvolta prototipul pentru software CP12. utilizează biblioteci de software

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/absolventul identifică și descrie concepte, principii și metode de bază din matematică, fizică, chimie, desen tehnic, economie și informatică.	Studentul/absolventul rezolvă probleme de matematică, fizică și chimie cu aplicabilitate în inginerie și validează soluția obținută.	Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer. Studentul/absolventul practică raționamentul logic, evaluarea și autoevaluare în luarea deciziilor.
Studentul/absolventul explică și interpretează rezultate teoretice și experimentale din matematică, fizică, chimie, economie, desen tehnic și informatică.	Studentul/absolventul achiziționează și prelucrează date, interpretează rezultate teoretice și experimentale.	Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate.

		Studentul/absolventul promovează dialogul, cooperarea, respectul față de ceilalți și interculturalitatea. Studentul/absolventul lucrează eficient ca membru în echipă sau lider al acesteia.
Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare, baze de date, inteligență artificială și inginerie software și modul lor de aplicare în probleme concrete	Studentul/absolventul alege şi explică concepte proprii specifice proiectării algoritmilor, programării orientate pe obiecte, programării logice și funcționale. Studentul/absolventul specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), aplicând elementele specifice ingineriei software.	Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei

6. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	Formarea unei gândiri algoritmice. Formarea și dezvoltarea deprinderilor de implementare a algoritmilor într-un limbaj de programare.

7. Conținutul predării și învățării

Aplicații (seminar / laborator / lucrări practice / proiect)	Nr. ore	Metode de predare	Observații
1. Prezentarea mediului de programare. Aplicaţii simple în Python. Tipuri de date, instrucțiunile ce implementează operațiile de bază.	2		
2. Instrucţiuni decizionale în Python	2		
3. Instrucțiuni repetitive în Python	2		
4. Algoritmi fundamentali în Python	2		
5. Aplicații cu tablouri unidimensionale în Python.	2		
6. Aplicații cu tablouri bidimensionale în Python	2		
7. Test laborator	1		
8. Aplicaţii cu caractere şi şiruri de caractere în Python	2		
9. Aplicaţii cu tuple, mulțimi, dicționare în Python	2		
10. Aplicaţii cu funcţii definite de utilizator în Python	4		
11. Aplicaţii cu fișiere în Python	4		
12. Programare modulară în Python	2		
13. Test laborator	1		

Bibliografie minimală recomandată

[1] Python Essential Reference, David Beazley
[2] Programming in Python 3: A Complete Introduction to the Python Language, Mark Summerfield
[3] Python: The Fundamentals Of Python Programming, Paul Jones, 2016
[4] Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Eric Matthes, 2015
[5] Python, https://www.python.org/
[6] B.W. Kernighan, D.M. Ritchie. Programarea in limbaj C. Teora, 2003
[7] B.Stroustrup, The C++ Programming Language, Third Edition by Bjarne Stroustrup, by AT&T. Addison Wesley Longman, Inc. ISBN 0201889544, 1997
[8] Pentiuc St. Gh., Turcu Cristina, Turcu Cornel, Mahalu George, Petrisor Cristinel - Programarea calculatoarelor, Universitatea “Stefan cel Mare” Suceava, 1995
[9] Stephen G. Kochan, Programming in C (4th Edition) (Developer's Library), Addison-Wesley Professional, ISBN-13: 978-0321776419, 2014
[10] L. Negrescu - Limbajele C și C++ pentru începători, vol. I, Editura Albastră, Cluj Napoca, 2012

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Evaluare finală	Criteriile generale de evaluare (corectitudinea cunoştinţelor, coerenţa	*Evaluare pe parcurs prin examinare scrisă* (prin scrierea şi punerea la	50

	logică, fluenţa în redactare)	punct a unui program funcţional pe calculator)	
Evaluare pe parcurs	Criterii specifice disciplinei (utilizarea corectă a metalimbajului specific domeniului limbajelor de programare)	*Evaluare continuă* (prin metode orale, test parţial şi probe practice)	50

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
23.09.2026		Ș.l. dr. inf. Laura-Bianca BILIUS

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
23.09.2026	Ș.l.dr.ing. Corneliu BUZDUGA
Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
24.09.2026	Conf.dr.ing. Eugen COCA

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
25.09.2026	Prof. univ. dr. ing. Laurențiu-Dan MILICI