

FIȘA DISCIPLINEI¹⁾

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiesti
1.2. Facultatea	Litere și Științe
1.3. Departamentul	Informatică, Tehnologia Informației, Matematică, Fizică
1.4. Domeniul de studii universitare	Informatică
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	Fundamentele Programarii
2.2. Titularul activităților de curs	Conf. dr. Simona Nicoară
2.3. Titularul activităților seminar/laborator	Lector dr. Aldescu Iuliana
2.4. Titularul activității proiect	-
2.5. Anul de studiu	I
2.6. Semestrul*	1
2.7. Tipul de evaluare	Examen
2.8. Categoria formativă** / regimul*** disciplinei	DF/O

*numărul semestrului este conform planului de învățământ;

**DF - Discipline fundamentale; DD - discipline de domeniu; DS - discipline de specialitate; DC - discipline complementare, DA - disciplina de aprofundare, DSI- disciplina de sinteza.

***obligatorie = O; opțională = A; facultativă = L

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

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. Seminar/laborator	2	3.4. Proiect	-
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. Seminar/laborator	28	3.8. Proiect	-
3.9. Distribuția fondului de timp							ore
Studiu după manual, suport de curs, bibliografie și notițe							30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							25
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							32
Tutoriat							4
Examinări							3
Alte activități							-
3.10 Total ore studiu individual	94						
3.11. Total ore pe semestru	150						
3.12. Numărul de credite	6						

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	
4.2. de competențe	-

¹⁾ Adaptare după Ordinul Ministrului educației, cercetării, tineretului și sportului nr. 5 703/2011 privind implementarea Codului național al calificărilor din învățământul superior, publicat în Monitorul Oficial al României, partea I, nr.880 bis / 13.XII.2011

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	➤ suport electronic la https://timf.upg-ploiesti.ro/cursuri/FP
5.2. de desfășurare a seminarului/laboratorului	➤ tehnica de calcul ➤ mediu de programare CodeBlocks

6. Competențe specifice acumulate

Competențe profesionale	<p>C1. PROGRAMAREA ÎN LIMBAJE DE NIVEL ÎNALT</p> <ul style="list-style-type: none"> • C1.1 DESCRIEREA ADECVATĂ A PARADIGMELOR DE PROGRAMARE ȘI A MECANISMELOR DE LIMBAJ SPECIFICE, PRECUM ȘI IDENTIFICAREA DIFERENȚEI DINTRE ASPECTELE DE ORDIN SEMANTIC ȘI SINTACTIC • C1.2 EXPLICAREA UNOR APLICAȚII SOFTWARE EXISTENTE, PE NIVELURI DE ABSTRACȚARE (ARHITECTURĂ, PACHETE, CLASE, METODE) UTILIZAND ÎN MOD ADECVAT CUNOȘTINȚELE DE BAZĂ • C1.3 ELABORAREA CODURILOR SURSĂ ADECVATE ȘI TESTAREA UNITARĂ A UNOR COMPONENTE ÎNTR-UN LIMBAJ DE PROGRAMARE CUNOSCUT, PE BAZA UNOR SPECIFICAȚII • C1.4 TESTAREA UNOR APLICAȚII PE BAZA UNOR PLANURI DE TEST • C1.5 DEZVOLTAREA DE UNITĂȚI DE PROGRAM ȘI ELABORAREA DOCUMENTAȚIILOR AFERENTE <p>C4. UTILIZAREA BAZELOR TEORETICE ALE INFORMATICII ȘI A MODELELOR FORMALE</p> <ul style="list-style-type: none"> • C4.1 DEFINIREA CONCEPTELOR ȘI PRINCIPIILOR DE BAZĂ ALE INFORMATICII, PRECUM ȘI A TEORIILOR ȘI MODELELOR MATEMATICE
Competențe transversale	<p>CT1. Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3. Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

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

7.1. Obiectivul general al disciplinei	➤ Însușirea cunoștințelor de bază necesare programării într-un limbaj de programare procedural
7.2. Obiectivele specifice	<p>După parcurgerea disciplinei studenții vor putea să:</p> <ul style="list-style-type: none"> ➤ definească și să identifice structurile algoritmice fundamentale necesare rezolvării unei probleme cu ajutorul calculatorului ➤ descrie în pseudocod și să codifice în limbajul de programare C algoritmul de rezolvare pentru problemele propuse în cadrul laboratorului ➤ aplice cunoștințele acumulate pentru rezolvarea informatică și a altor probleme, cu un grad de dificultate asemănător cu cele abordate la laborator ➤ recunoască specificul paradigmei de programare procedurală comparativ cu alte paradigme

8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1.Concepte de bază. Dată, informație, cunoștințe. Paradigme	4		

de programare. Conceptul de programare structurată. Algoritmi-caracteristici, descriere. Teorema de structură. Reprezentarea structurilor algoritmice fundamentale. Proceduri si functii – concepte. Translatarea, linkeditarea și încărcarea programelor pentru execuție. Generații de limbaje de programare		Expunere pe baza suportului de curs, studii de caz, conversație, dezbateri	
2.Structura generală a unui program C. Directive preprocesor. Elemente de bază ale limbajului C. Tipuri de date fundamentale în C, variabile, constante. Functii uzuale de intrare-iesire pentru consolă. Descriptori de format. Operatori în C. Clase de precedență. Conversii de tip explicite si implicite	4		
3.Instrucțiuni de control ale programului in limbajul C (instrucțiuni expresie, instrucțiuni de decizie, instrucțiuni iterative, instrucțiuni de salt)	2		
4.Tablouri, inițializarea tablourilor, șiruri de caractere. Functii uzuale pentru prelucrarea șirurilor de caractere	2		
5..Pointeri - declarare și inițializare. Pointeri și tablouri unidimensionale. Pointeri către tipul char și șiruri de caractere. Pointeri către tablouri unidimensionale. Operații aritmetice cu pointeri. Pointeri dubli și tablouri bidimensionale	2		
6. Structuri. Uniuni. Campuri de biti, enumerări, declarații typedef	4		
7. Functii utilizator - declarație și definiție. Transferul datelor către functii (prin valoare și prin referință). Argumentele funcției main(). Pointeri la functii. Functii recursive	4		
8. Clase de memorare	2		
9. Functii pentru prelucrarea fisierelor (functii pentru deschidere si inchidere fisier - fopen() si fclose()). Functii pentru citirea si scrierea unui caracter - fgetc() si fputc(). Functii pentru transferul șirurilor de caractere - fgets() si fputs(). Functii pentru transfer cu format - fscanf() si printf(). Functii pentru transferul blocurilor de date - fread() si fwrite(). Functiile feof() si rewind(). Functii pentru citirea si modificarea indicatorului de pozitie al fisierului - fseek(), fgetpos(), fsetpos(), ftell())	4		
Bibliografie <ol style="list-style-type: none"> 1. J. Glenn Brookshear, <i>Introducere in Informatica</i>, Editura Teora, Bucuresti, 1999 2. Cristian Marinoiu, <i>Programarea in limbajul C</i>, Editura Universitatii din Ploiesti, Ploiesti, 2000 3. Cristian Marinoiu, Elena Simona Nicoară, <i>Introducere în Informatică</i>, Editura Universitatii din Ploiesti, Ploiesti, 2014 4. Brian Kernighan, Dennis Ritchie, <i>The C programming Language</i>, Prentice Hall, 1988, disponibilă la http://www.ime.usp.br/~pf/Kernighan-Ritchie/C-Programming-Ebook.pdf 			

5., Herbert Schildt, C: <i>Manual complet</i> , Editura Teora, Bucuresti, 1997			
8.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
1.Exemple de algoritmi; reprezentarea lor în pseudocod și schemă logică	2	Expunere, studii de caz, conversație, dezbateri, aplicații pe calculator	
2.Programe în limbajul C care utilizează instrucțiuni de intrare/iesire, instrucțiuni de control (expresie, de decizie, iterative și de salt). Test de evaluare	6		
3.Programe în limbajul C care utilizează tablouri. Test de evaluare	8		
4.Programe care utilizează structuri, declarații typedef, pointeri, fișiere și funcții utilizator. Test de evaluare	10		
5.Examinare orală cu bilete: aplicație pe calculator	2		
Bibliografie			
<ol style="list-style-type: none"> 1. Cristian Marinoiu, <i>Programarea în limbajul C</i>, Editura Universitatii din Ploiesti, Ploiesti, 2000 2. Cristian Marinoiu, Elena Simona Nicoară, <i>Introducere în Informatică</i>, Editura Universitatii din Ploiesti, Ploiesti, 2014 3. Brian Kernighan, Dennis Ritchie, <i>The C programming Language</i>, Prentice Hall, 1988, disponibilă la http://www.ime.usp.br/~pf/Kernighan-Ritchie/C-Programming-Ebook.pdf 4. Herbert Schildt, C: <i>Manual complet</i>, Editura Teora, Bucuresti, 1997 			

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

Cursul și laboratorul sunt astfel concepute încât, prin competențele formate, să răspundă cerințelor pieței muncii.

Disciplina există în planul de învățământ al specializărilor de Informatică din universitățile românești

10.Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Calitatea prezentării și a răspunsurilor la examinarea finală	Examinare orală cu bilete	25%
10.5. Seminar/laborator	Calitatea activității desfășurate în cadrul laboratorului	Intrebări privitoare la modul de rezolvare a problemelor propuse în cadrul laboratorului	25%
	Calitatea răspunsurilor la testele de evaluare	Teste de evaluare pe parcursul semestrului	25%
	Calitatea rezolvării pe	Examinare orală cu bilete:	

	calculator a aplicației la examinarea finală	aplicație pe calculator	25%
10.6. Standard minim de performanță			
➤ Implementarea și comentarea de unități de program în limbajul de programare C și folosirea eficientă a mediului de programare CodeBlocks			

Data
completării
15 sept. 2021

Semnătura titularului de
curs

Semnătura titularului de laborator

Data avizării în
departament
16.09.2021

Director de departament
Conf. dr. Gabriela Moise

Decan
Prof. univ. dr. Mihaela Suditu