

# FIȘA DISCIPLINEI<sup>1)</sup>

## 1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiesti
1.2. Facultatea	Litere si Stiințe
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 Programării
2.2. Titularul activităților de curs	Conf.dr. mat. Marinoiu Cristian
2.3. Titularul activităților aplicative	Lector dr. Dobre Iuliana, Lector dr.Schiopu Daniela, Lector dr. Dragomir Elia
2.4. Anul de studiu	I
2.5. Semestrul *	1
2.6. Tipul de evaluare	Examen
2.7. Categoria formativă** / regimul*** disciplinei	F0/O

\* numărul semestrului este conform planului de învățământ;

\*\* fundamentală = F0; de domeniu = D1; de specialitate = S2; complementară = C3

\*\*\* 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. Total ore din planul de învățământ	56	din care: 3.5. curs	28	3.6. Seminar/laborator	28
3.7. Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și notițe					29
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					14
Examinări					3
Alte activități					-
3.7. Total ore studiu individual	94				
3.8. Total ore pe semestru	150				
3.9. 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	<ul style="list-style-type: none"> <li>• Sală cu dotare clasică și mijloace de proiecție</li> </ul>
5.2. de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>• Sală dotată cu calculatoare și software adecvat</li> </ul>

## 6. Competențe specifice acumulate

<b>Competențe profesionale</b>	<p>C1.1 Descrierea adecvată a paradigmelor de programare și a mecanismelor de limbaj specifice</p> <p>C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații</p> <p>C1.4 Testarea unor aplicații pe baza unor planuri de test.</p> <p>C4.1 Definirea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice</p>
<b>Competențe transversale</b>	<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>

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

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

## 8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
<p>1. Generații de limbaje de programare. Paradigme de programare. Conceptul de programare structurată Algoritmi-caracteristici, descriere. Teorema de structură .Reprezentarea structurilor algoritmice fundamentale. Tipuri de date si operații elementare. Structuri de date. Proceduri si functii (pseudocod și schemă logică) Translatarea, linkeditarea și încărcarea programelor pentru execuție.</p> <p>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</p> <p>3. Instrucțiuni de control ale programului (instrucțiuni expresie, instrucțiuni de decizie, instrucțiuni iterative, instrucțiuni de salt).</p> <p>4. Tablouri, inițializarea tablourilor, siruri de caractere . Funcții uzuale pentru prelucrarea șirurilor de caractere.</p> <p>5. Pointeri-declarație 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.</p> <p>6. Structuri. Uniuni. Campuri de biti, enumerări, declarații typedef.</p> <p>7. Functii utilizator- declarație și definiție. Transferul datelor către funcții (prin valoare și prin referință). Argumentele funcției main(). Pointeri la funcții. Funcții recursive.</p> <p>8. Clase de memorare</p> <p>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 sirurilor de caractere-fgets() si fputs(). Functii pentru transfer cu format-fscanf() si fprintf(). 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()).</p>	<p>1. 4 2. 4 3. 2 4. 2 5. 2 6. 4 7. 4 8. 2 9. 4</p>	<p>Expunere, studii de caz, conversație, dezbateri</p>	

Bibliografie			
<ol style="list-style-type: none"> <li>1. Brookshear J. Glenn, Introducere in Informatica, Editura Teora, Bucuresti, 1999</li> <li>2. Cristian Marinoiu, Programarea in limbajul C, Editura Universitatii din Ploiesti, Ploiesti, 2000</li> <li>3. Cristian Marinoiu, Elena Simona Nicoară, Introducere în Informatică, Editura Universitatii din Ploiesti, Ploiesti, 2014</li> <li>4. Brian Kernighan, Dennis Ritchie, The C programming Language, Prentice Hall, 1988, disponibilă la <a href="http://www.ime.usp.br/~pf/Kernighan-Ritchie/C-Programming-Ebook.pdf">http://www.ime.usp.br/~pf/Kernighan-Ritchie/C-Programming-Ebook.pdf</a></li> <li>5. Herbert Schildt C: Manual complet, Editura Teora, Bucuresti, 1997</li> </ol>			
<b>8.2. Seminar / laborator/proiect</b>	Nr. ore	Metode de predare	Observații
1.Exemple de algoritmi; reprezentarea lor în pseudocod si schemă logică	1.2	Expunere, studii de caz, conversație, dezbateri: aplicații pe calculator	
2.Programe in limbajul C care utilizează instrucțiuni de intrare/iesire, instrucțiuni de decizie și instrucțiuni iterative și tablouri de date	2.10		
3. Programe care utilizează structuri. uniuni.campuri de biti, enumerări, declarații typedef și fișiere	3.16		
Bibliografie			
<ol style="list-style-type: none"> <li>1. Brookshear J. Glenn, Introducere in Informatica, Editura Teora, Bucuresti, 1999</li> <li>2. Cristian Marinoiu, Programarea in limbajul C, Editura Universitatii din Ploiesti, Ploiesti, 2000</li> <li>3. Cristian Marinoiu, Elena Simona Nicoară, Introducere în Informatică, Editura Universitatii din Ploiesti, Ploiesti, 2014</li> <li>4. Brian Kernighan, Dennis Ritchie, The C programming Language, Prentice Hall, 1988, disponibilă la <a href="http://www.ime.usp.br/~pf/Kernighan-Ritchie/C-Programming-Ebook.pdf">http://www.ime.usp.br/~pf/Kernighan-Ritchie/C-Programming-Ebook.pdf</a></li> <li>5. Herbert Schildt C: Manual complet, Editura Teora, Bucuresti, 1997</li> </ol>			

### 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 Programarea procedurală 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/proiect	Calitatea activității desfășurate în cadrul laboratorului	Intrebari privitoare la modul de rezolvare a problemelor propuse in cadrul laboratorului	25%
	Calitatea răspunsurilor la testele de control	Teste de control pe parcursul semestrului	25%
	Calitatea rezolvării pe calculator a aplicației la examinarea finală	Examinare orală cu bilete: aplicație pe calculator	25%

10.6. Standard minim de performanță

Să descrie corect și să exemplifice structurile algoritmice fundamentale în limbajul C, să poată elabora și implementa algoritmi care necesită structuri de decizie și de iterație, să utilizeze funcții pentru implementarea unui program în C

Data completării  
15.09.2018

Semnătura titularului de curs  
Conf. dr. Marinoiu Cristian

Semnătura titularului de seminar/laborator  
Lector dr. Dobre Iuliana, Lector  
dr. Schiopolu Daniela, Lector dr. Dragomir  
Elia

Data avizării în departament

---

Semnătura directorului de departament

Conf. dr. Gabriela Moise