

# FIȘA DISCIPLINEI<sup>1)</sup>

## 1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Litere și Științe
1.3. Departamentul	Informatică, Tehnologia Informației, Matematică și 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	Programare orientată pe obiecte
2.2. Titularul activităților de curs	Șef lucr. dr. ing. Ioniță Irina
2.3. Titularul activităților aplicative	Lect. dr. mat. Dragomir Elia Georgiana Șef lucr. dr. ing. Ioniță Irina
2.4. Anul de studiu	I
2.5. Semestrul *	II
2.6. Tipul de evaluare	E
2.7. Categoria formativă** / regimul*** disciplinei	DS/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. 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					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					40
Tutoriat					6
Examinări					8
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	➤ Fundamentele programării ➤ Algoritmi fundamentali
4.2. de competențe	➤ Cunoașterea unui limbaj de programare (limbajul C) și a etapelor de dezvoltare

<sup>1)</sup> 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

a aplicațiilor cu acest limbaj de programare

## 5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	➤ Cursul se desfășoară într-o sală dotată cu calculator, videoproiector, acces Internet, tablă.
5.2. de desfășurare a seminarului/laboratorului	➤ Laboratorul se desfășoară în săli de laborator, aplicațiile fiind implementate pe stații de lucru într-un mediu de programare pentru limbajul C++ (Code Blocks).

## 6. Competențe specifice acumulate

<b>Competențe profesionale</b>	C.1.1. Descrierea adecvată a paradigmei de programare orientată pe obiecte și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic C.1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază C.1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente în limbajul C++, pe baza unor specificații C.1.4 Testarea unor aplicații orientate-obiect pe baza unor planuri de test. C.1.5 Dezvoltarea unor aplicații orientate obiect și elaborarea documentațiilor aferente.
<b>Competențe transversale</b>	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 CT2. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatică de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse

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

7.1. Obiectivul general al disciplinei	➤ Formarea de competențe profesionale și transversale necesare obținerii calificării. Obiectivul principal al disciplinei constă în familiarizarea studenților cu abordarea orientată-obiect și deprinderea acestora cu un limbaj de programare de nivel înalt (C++) pentru dezvoltarea de aplicații cu aplicabilitate în domeniul de specialitate.
7.2. Obiectivele specifice	Formarea următoarelor competențelor profesionale ( C1.1 –C1.5) și transversale (CT1, CT2). La sfârșitul cursului, studenții vor fi capabili să: ➤ descrie adecvat paradigma programării orientate pe obiecte prin definirea conceptele de bază și să realizeze o comparație cu celelalte tipuri de programări (ex. programare procedurală); ➤ explice nevoia utilizării programării orientate pe obiecte, identificând avantajele acesteia; ➤ deprindă un nou mod de programare (orientat-obiect) și să învețe un limbaj orientat-obiect (C++); ➤ proiecteze și să implementeze aplicații în limbajul orientat obiect

	<p>studiat (C++) cu aplicabilitate în domeniul de specialitate;</p> <ul style="list-style-type: none"> <li>➤ utilizeze UML pentru analiza și proiectarea aplicațiilor orientate-obiect;</li> <li>➤ aplice reguli de muncă organizată și eficientă, să manifeste atitudini responsabile față de domeniul didactic-științific, să respecte principiile și norme de etică profesională, să lucreze în echipă la elaborarea de proiecte de complexitate medie.</li> </ul>
--	---

## 8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
Nevoia utilizării programării orientate obiect în dezvoltarea aplicațiilor software	2	Prelegere, susținută de prezentări PPT, conversații, explicații, exemplificări practice la tablă, studii de caz	
UML – limbaj orientat obiect	2		
Analiza și proiectarea orientată obiect a sistemelor software	2		
Concepte de bază ale programării orientate pe obiecte	2		
Clase și obiecte. Funcțiile constructor și destructor. Funcții friend. Funcții inline	4		
Membri statici. Tablouri de obiecte	4		
Redefinirea operatorilor	2		
Moștenirea în programarea orientată pe obiecte. Clase de bază și clase derivate. Moștenirea multiplă	4		
Funcții virtuale și polimorfism	2		
Stream-uri de intrare/ieșire	4		

### Bibliografie

1. T. Budd (1997), *An Introduction to Object-Oriented Programming*, Addison Wesley.
2. H. M. Deitel, P. J. Deitel (2001), *C++ How to program*, Prentice Hall.
3. F. Drouillon (2014), *Du C au C++. De la programmation procedurale a l'objet, 2ieme edition*, ENI Edition.
4. B. Eckel (2000), *Thinking in C++, 2nd Edition*, Prentice Hall, <http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>
5. S. Guțu, L. Dumitrașcu, G. Marcu, L. Ioniță (2005), *Analiza și proiectarea orientată obiect a sistemelor informatice cu UML*, INDE Ploiești, Editura Universității din Ploiești, ISBN 973-719-012-2.
6. L. Ioniță, et al (2015), *Diagramele UML 2. Dicționar. Studii de caz. Aplicație Web*, Editura Universității Petrol-Gaze din Ploiești, Ploiești.
7. R. Lafore (2001), *Object-Oriented Programming in C++ (4th Edition)*, SAMS.
8. M. Oprea (2003), *Programare orientată pe obiecte – Exemple în limbajul C++*, Editura MATRIX ROM, București.
9. M. Preda, A. Mircea, D. Preda, C. Teodorescu (2010), *Introducere în programarea orientată-obiect. Concepte fundamentale din perspectiva ingineriei software*, Editura Polirom, București.
10. S. Reiss (1999), *A practical introduction to software design with C++*, John Wiley & Sons.
11. D. Schiopu, I. Ionita, L. Ionita, G. Marcu, L. Dumitrascu (coordonator) (2016), *De la C la C++, C#. Exerciții didactice*, Editura Universității Petrol-Gaze din Ploiești, Ploiești.
12. C. Spircu, I. Lopătan (1995), *Analiza, proiectarea și programarea orientate spre obiecte*, Editura Teora,

Bucuresti.			
B. Stroustrup (1986), <i>The C++ Programming Language</i> , Addison Wesley.			
8.2. Seminar / laborator/proiect	Nr. ore	Metode de predare	Observații
Recapitulare limbajul C	4	Prelegere, discuții, rezolvarea la calculator a unor probleme propuse și explicații la tablă	
Abstractizarea datelor. Trecerea de la tipul struct la clase	2		
Clase și obiecte. Constructori. Destructor	4		
Redefinirea operatorilor	4		
Clase de bază și clase derivate	4		
Funcții virtuale și polimorfism	2		
Fișiere	2		
Analiza, proiectarea și implementarea OO a unei aplicații cu UML. Trecerea de la diagrama cazurilor de utilizare la diagrama claselor conceptuale	2		
UML pentru analiza și proiectarea aplicațiilor software. VP for UML. Prezentare generală. Realizarea diagramelor în UML. Studii de caz. Discuție proiecte	2		
Prezentarea proiectelor	2	Dezbaterea temelor	
Bibliografie			
1. F. Drouillon (2014), <i>Du C au C++</i> . De la programmation procedurale a l'object, 2ieme edition, ENI Edition.			
2. R. Lafore (2001), <i>Object-Oriented Programming in C++ (4th Edition)</i> , SAMS.			
3. M. Oprea (2003), <i>Programare orientată pe obiecte – Exemple în limbajul C++</i> , Editura MATRIX ROM, București.			
4. D. Schiopu, I. Ionita, L. Ionita, G. Marcu, L. Dumitrascu (coordonator) (2016), <i>De la C la C++, C#. Exerciții didactice</i> , Editura Universității Petrol-Gaze din Ploiești, Ploiești.			
C. Spircu, I. Lopătan (1995), <i>Analiza, proiectarea și programarea orientate spre obiecte</i> , Editura Teora, Bucuresti.			

## 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 *Programare orientată pe obiecte* respectă recomandările IEEE și ACM legate de Curricula pentru specializarea Informatică. Noțiunile teoretice prezentate în curs și laborator sunt conform standardelor Internet Engineering Task Force.
- Conținuturile disciplinei sunt specifice domeniului dezvoltării aplicațiilor software orientate-obiect, temele de lucru individual și proiectele, precum și lista de probleme aferentă fiecărui capitol din curs sunt astfel alese încât să corespundă tipurilor de activitate specifice specializării.

## 10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Dobândirea de cunoștințe de bază privind abordarea orientată-obiect	Test grilă	30%
	Înșușirea unui limbaj minim de specialitate, specific domeniului		
10.5. Seminar/laborator/proiect	Înșușirea tehnicilor de lucru în dezvoltarea de aplicații orientate-obiect specifice domeniului	Teste parțiale de verificare a cunoștințelor acumulate și a stilului de programare ce constau în rezolvarea unor probleme pe calculator	30%
		Examinarea proiectelor realizate de echipe formate din maxim trei studenți, având o temă la alegere, referitoare la dezvoltarea unei aplicații orientate obiect	30%
	Participarea activă la orele de laborator	Întrebări pe parcursul semestrului	10%
10.6. Standard minim de performanță			
<ul style="list-style-type: none"> <li>➤ declararea unei clase (cu cele trei secțiuni: private, public și protected); declararea și definirea a două funcții membru (un constructor și o funcție la alegere); declararea a două obiecte ce aparțin clasei respective și apelarea unei funcții.</li> <li>➤ obținerea notei minim 5 la toate cele trei metode de evaluare (test grilă, teste parțiale și proiect).</li> </ul>			

Data completării  
01.10.2018

Semnătura titularului de curs

Semnătura titularului de seminar/laborator

Data avizării în departament

Semnătura directorului de departament