

FIȘA DISCIPLINEI¹⁾

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. Ciclu 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. Irina Ioniță
2.3. Titularul activităților aplicative	lect.mat.dr. Elia Georgiana Dragomir
2.4. Anul de studiu	2
2.5. Semestrul *	3
2.6. Tipul de evaluare	E
2.7. Categoria formativă** / regimul*** disciplinei	S2/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	30				
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	10				
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	20				
Tutoriat	6				
Examinări	3				
Alte activități					
3.7. Total ore studiu individual	69				
3.8. Total ore pe semestru	125				
3.9. Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none">➤ Programare procedurală➤ Programare procedurală avansată
4.2. de competențe	<ul style="list-style-type: none">➤ Cunoașterea unui limbaj de programare (limbajul C) și a etapelor de dezvoltare a aplicațiilor cu acest limbaj de programare

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	➤ Cursul se desfășoară prin utilizarea tehnologiilor multimedia (videoproiector), cu prezentarea în direct a noțiunilor referitoare la programarea orientată-obiect și cu participarea activă a studenților
5.2. de desfășurare a seminarului/laboratorului	➤ Laboratorul se desfășoară în săli de laborator, aplicațiile fiind implementate pe stații de lucru pe care rulează compilatoare de C

6. Competențe specifice acumulate

Competențe profesionale	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 C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente în limbajul C++, pe baza unor specificații C1.4 Testarea unor aplicații orientate-obiect pe baza unor planuri de test. C1.5 Dezvoltarea unor aplicații orientate obiect și elaborarea documentațiilor aferente.
Competențe transversale	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++);

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

8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
Nevoia utilizării programării orientate obiect în dezvoltarea aplicațiilor software	2	Prelegere	
UML – limbaj orientat obiect	2	Prelegere	
Analiza și proiectarea orientată obiect a sistemelor software	2	Prelegere	
Concepte de bază ale programării orientate pe obiecte	2	Prelegere	
Clase și obiecte. Funcțiile constructor și destructor. Funcții friend. Funcții inline	4	Prelegere și demonstrații practice la tablă	
Membri statici. Tablouri de obiecte	4	Prelegere și demonstrații practice la tablă	
Redefinirea operatorilor	2	Prelegere și demonstrații practice la tablă	
Moștenirea în programarea orientată pe obiecte. Clase de bază și clase derivate. Moștenirea multiplă	4	Prelegere și demonstrații practice la tablă	
Funcții virtuale și polimorfism	4	Prelegere și demonstrații practice la tablă	
Stream-uri de intrare/ieșire	2	Prelegere și demonstrații practice la tablă	
Bibliografie <ol style="list-style-type: none"> 1. T. Budd (1997), <i>An Introduction to Object-Oriented Programming</i>, Addison Wesley. 2. H. M. Deitel, P. J. Deitel (2001), <i>C++ How to program</i>, Prentice Hall. 3. F. Drouillon (2014), <i>Du C au C++. De la programmation procedurale a l'object</i>, 2ieme edition, ENI Edition. 4. B. Eckel (2000), <i>Thinking in C++, 2nd Edition</i>, Prentice Hall, http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html 5. S. Guțu, L. Dumitrașcu, G. Marcu, L. Ioniță (2005), <i>Analiza și proiectarea orientată obiect a sistemelor informatice cu UML</i>, INDE Ploiești, Editura Universității din Ploiești, ISBN 973-719-012-2. 6. L. Ioniță, et al (2015), <i>Diagramele UML 2. Dicționar. Studii de caz. Aplicație Web</i>, Editura Universității Petrol-Gaze din Ploiești, Ploiești. 7. R. Lafore (2001), <i>Object-Oriented Programming in C++ (4th Edition)</i>, SAMS. 8. M. Oprea (2003), <i>Programare orientată pe obiecte – Exemple în limbajul C++</i>, 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.
13. B. Stroustrup (1986), *The C++ Programming Language*, Addison Wesley.

8.2. Seminar / laborator/proiect	Nr. ore	Metode de predare	Observații
Recapitulare limbajul C	4	Lucrul la calculator și explicații la tablă	
Abstractizarea datelor. Trecerea de la tipul struct la clase	2	Lucrul la calculator și explicații la tablă	
Clase și obiecte. Constructori. Destructor	4	Lucrul la calculator și explicații la tablă	
Redefinirea operatorilor	4	Lucrul la calculator și explicații la tablă	
Clase de bază și clase derivate	4	Lucrul la calculator și explicații la tablă	
Funcții virtuale și polimorfism	2	Lucrul la calculator și explicații la tablă	
Fișiere	2	Lucrul la calculator și explicații la tablă	
Analiza, proiectarea și implementarea OO a unei aplicații cu UML. Trecerea de la diagrama cazurilor de utilizare la diagrama claselor conceptuale	2	Lucrul la calculator și explicații la tablă	
UML pentru analiza și proiectarea aplicațiilor software. VP for UML. Prezentare generală. Realizarea diagramelor în UML. Studii de caz. Discuție proiecte	2	Lucrul la calculator, explicații la tablă, discuții	
Prezentarea proiectelor	2	Dezbatere	
Bibliografie			
<ol style="list-style-type: none"> 1. F. Drouillon (2014), <i>Du C au C++. De la programmation procedurale a l'object, 2ieme edition</i>, 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. 5. 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

<ul style="list-style-type: none"> ➤ Cursul și laboratorul sunt astfel concepute încât, prin competențele formate, să răspundă cerințelor pieței muncii. ➤ Disciplina <i>Programare orientată pe obiecte</i> 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 paradigma programării orientate-obiect reflectată prin calitatea răspunsurilor la examen, coerența argumentării și capacitatea de sinteză	Examen scris ce constă din tratarea a două subiecte de teorie	30%
	Însușirea unui limbaj de programare de specialitate (limbajul C++)		
10.5. Seminar/laborator/proiect	Însușirea tehnicilor de lucru în dezvoltarea de aplicații orientate-obiect specifice domeniului	Evaluarea unui proiect realizat de o echipă formată din maxim 3 studenți, cu temă la alegere – dezvoltarea unei aplicații orientate obiect	70%
10.6. Standard minim de performanță			
<ul style="list-style-type: none"> ➤ 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. 			

Data completării
28.09.2016

Semnătura titularului de curs
Șef.lucr.dr.ing. Ioniță Irina

Semnătura titularului de seminar/laborator
Lect. mat. dr. Dragomir Elia Georgiana

Data avizării în departament

Semnătura directorului de departament,,