

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. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	Tehnici avansate de programare
2.2. Titularul activităților de curs	Lector dr. Daniela Tudorică
2.3. Titularul activităților aplicative	Lector dr. Daniela Tudorică
2.4. Anul de studiu	II
2.5. Semestrul *	4
2.6. Tipul de evaluare	Examen
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:	2	3.3. Seminar/laborator	2
		3.2. curs			
3.4. Total ore din planul de învățământ	56	din care:	28	3.6. Seminar/laborator	28
		3.5. curs			
3.7. Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și notițe					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					5
Tutoriat					5
Examinări					4
Alte activități					0
3.7. Total ore studiu individual	44				
3.8. Total ore pe semestru	100				
3.9. Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none">Programare procedurală, Algoritmi și structuri de date, Programare orientată pe obiecte
--------------------	---

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

4.2. de competențe	<ul style="list-style-type: none"> Proiectarea algoritmilor, Cunoștințe de programare în limbajul C/C++
--------------------	--

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	<ul style="list-style-type: none"> sală de curs multimedia pentru realizarea de prelegeri, conversații, dezbateri
5.2. de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> laborator dotat cu sisteme de calcul cu mediu vizual de programare instalat (Microsoft Visual C#)

6. Competențe specifice acumulate

Competențe profesionale	<p>C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstracție (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază.</p> <p>C2.1 Identificarea de metodologii adecvate de dezvoltare a sistemelor software.</p> <p>C2.2 Identificarea și explicarea mecanismelor adecvate de specificare a sistemelor software.</p> <p>C3.2 Identificarea și explicarea modelelor informatice de bază adecvate domeniului de aplicare.</p> <p>C2.3 Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice.</p> <p>C3.3 Utilizarea modelelor și instrumentelor informatice și matematice pentru rezolvarea problemelor specifice domeniului de aplicare.</p>
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>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</p>

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 fundamentele teoretice și aplicative ale tehnicilor avansate de programare a calculatoarelor.
7.2. Obiectivele specifice	<p>Formarea competențelor profesionale (C1.2, C2.1, C2.2, C3.2, C2.3, C3.3) și transversale (CT1, CT2).</p> <p>La finalul activităților, studentul va fi capabil să:</p> <ul style="list-style-type: none"> utilizeze un mediu vizual de programare; abordeze rezolvarea diverselor categorii de probleme prin tehnici de programare adecvate; folosească abilitatea de analiză a complexității algoritmilor; 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.

8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1. Prezentarea obiectivelor disciplinei, a modului de desfășurare a orelor de curs și de laborator, prezentarea modului de evaluare. Recapitulare: etapele rezolvării unei probleme cu ajutorul calculatorului, algoritmi.	1. 2 ore 2. 2 ore 3. 2 ore 4. 2 ore 5. 2 ore 6. 2 ore	Expunere, studii de caz, conversație, dezbateri, utilizare tehnologii multimedia	
2. Algoritmi. Elemente de analiză a complexității algoritmilor.	7. 2 ore 8. 2 ore		
3. Tehnici clasice de programare. Tehnica forței brute. Tehnica reducerii dimensiunii problemei. Recursivitatea ca tehnică de programare. Analiza complexității algoritmilor recursivi.	9. 2 ore 10. 4 ore 11. 2 ore 12. 3 ore 13. 1 oră		
4. Tehnica de programare Greedy. Descrierea metodei, exemple, analiza complexității.			
5. Metoda programării dinamice. Descrierea metodei, exemple, analiza complexității.			
6. Metoda Backtracking. Descrierea metodei, exemple, analiza complexității.			
7. Metoda Divide et Impera. Descrierea metodei, exemple, analiza complexității.			
8. Metoda Branch and Bound. Descrierea metodei, exemple, analiza complexității.			
9. Metode de sortare și căutare în vectori. Analiză comparativă			
10. Elemente avansate de programare vizuală (Microsoft Visual C#). Programare generică. Tratarea excepțiilor.			
11. Accesarea și prelucrarea datelor din baze de date.			
12. Elemente de grafică în C#.			
13. Recapitulare.			
Bibliografie			
1. Cristea V., Athanasiu I., Kalisz E., Iorga V., <i>Tehnici de programare</i> , Ed. Teora, București, 1993			
2. Coșulschi M., <i>Algoritmi fundamentali. Proiectare și implementare</i> , Ed. Universitaria, Craiova, 2015			
3. Knuth D.E., <i>The art of computer programming, Vol. I – Fundamental Algorithms</i> , ediția a treia, Addison Wesley Longman, 1997			
4. Knuth D.E., <i>The art of computer programming, Vol. III – Sorting and Searching</i> , ediția a doua, Addison Wesley Longman, 1998			
5. Levitin A., <i>Introduction to the design and analysis of algorithms</i> , Pearson Education, third edition, 2012			
6. Livovschi L., Georgescu H. <i>Sinteza și analiza algoritmilor</i> , Universitatea din București, Fac. de Matematică, București, 1985			
7. Odagescu I., <i>Tehnici de programare</i> , Ed. Intact, București, 1994			
8. Robert Sedgewick, <i>Algorithms</i> , Brown University, Addison-Wesley Publishing Company, Addison-Wesley Series in Computer Science 1983			
9. Zaharie D., <i>Introducere în proiectarea și analiza algoritmilor</i> , Ed. Eubeea, 2008			

8.2. Seminar / laborator/proiect	Nr. ore	Metode de predare	Observații
1. Familiarizarea cu mediul de programare utilizat (Microsoft Visual C#). Realizarea de aplicații Windows – GUI, componente, proprietăți, evenimente, tratarea excepțiilor.	1. 2 ore 2. 2 ore 3. 2 ore 4. 4 ore 5. 2 ore	Dezbateri, studii de caz, exemple, angajarea fiecărui student în descoperirea cunoștințelor.	
2. Analiza empirică a complexității unui algoritm (calculul timpului de execuție).	6. 4 ore 7. 2 ore		
3. Aplicații care utilizează recursivitatea.	8. 2 ore		
4. Tehnica de programare Greedy. Aplicații.	9. 2 ore		
5. Metoda programării dinamice. Aplicații.	10. 2 ore		
6. Metoda Backtracking. Aplicații.	11. 2 ore		
7. Metoda Divide et Impera. Aplicații.	12. 2 ore		
8. Metoda Branch and Bound. Aplicații.			
9. Funcții și clase generice (C3)			
10. Accesarea și prelucrarea datelor din baze de date (C#)			
11. Grafică în C#			
12. Evaluare laborator			
Bibliografie			
1. Coșulski M., Algoritmi fundamentali. Proiectare și implementare, Ed. Universitaria, Craiova, 2015			
2. Knuth D.E., <i>The art of computer programming, Vol. I – Fundamental Algorithms</i> , ediția a treia, Addison Wesley Longman, 1997			
3. Knuth D.E., <i>The art of computer programming, Vol. III – Sorting and Searching</i> , ediția a doua, Addison Wesley Longman, 1998			
4. Levitin A., <i>Introduction to the design and analysis of algorithms</i> , Pearson Education, third edition, 2012			
5. Zaharie D., <i>Introducere în proiectarea și analiza algoritmilor</i> , Ed. Eubeea, 2008			
6. Jon Skeet, <i>C# in Depth</i> , Manning Publications, 2008			
7. <i>C# Programming Guide</i> , https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx			
8. <i>Tutorial C#</i> , http://www.tutorialspoint.com/csharp/index.htm			

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.
- Conținutul acestei discipline oferă cursanților cunoștințe avansate de proiectare și implementare a aplicațiilor, cunoștințe indispensabile în cazul în care absolventul decide să urmeze o carieră în domeniul software (capacitatea de rezolvare a diverselor categorii de probleme prin tehnici de programare adecvate și abilitatea de analiză a complexității algoritmilor).

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Completitudinea și corectitudinea cunoștințelor,	Examen scris	50%

	capacitatea de a opera cu cunoștințele asimilate, capacitatea de analiză, de interpretare personală, gradul de asimilare a limbajului de specialitate și capacitatea de comunicare.		Pentru promovarea examenului minim nota 5 la proba scrisă.
10.5. Seminar/laborator/ proiect	Aplicarea practică a elementelor prezentate la orele de curs și de laborator, prin realizarea de aplicații folosind tehnici avansate de programare.	Evaluarea temei de casă	50% Pentru promovarea examenului minim nota 5 la laborator.
10.6. Standard minim de performanță			
Asimilarea limbajului de specialitate privind tehnici avansate de programare. Abilitatea de aplicare practică a cunoștințelor privind tehnicile avansate de programare. Realizarea și întreținerea unor aplicații informatice pentru rezolvarea unor probleme reale de complexitate medie.			

Data completării
29.09.2020

Semnătura titularului de curs
Lector dr. Daniela Tudorică

Semnătura titularului de seminar/laborator
Lector dr. Daniela Tudorică

Data avizării în departament

Semnătura directorului de departament
Conf. dr. Gabriela Moise