

FI A 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	Informatic, Tehnologia Informației, Matematică și Fizică
1.4. Domeniul de studii universitare	Informatic
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Informatic

2. Date despre disciplină

2.1. Denumirea disciplinei	Algoritmi și Structuri de Date
2.2. Titularul activităților de curs	Conf. dr. ing. Monica Vladoiu
2.3. Titularul activităților aplicative	Lector dr. Daniela Schiopu
2.4. Anul de studiu	II
2.5. Semestrul*	3
2.6. Tipul de evaluare	E
2.7. Categoria formativ ** / regimul*** disciplinei	F0/O

*numărul semestrului este conform planului de învățământ;

** fundamental = F0; de domeniu = D1; de specialitate = S2; complementar = C3

***obligatorie = O; opțional = A; facultativ = 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	70	din care: 3.5. curs	28	3.6. Seminar/laborator	42
3.7. Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătirea seminariilor/laboratoare, teme, referate, portofolii și eseuri					40
Tutoriat					-
Examinări					-
Alte activități					-
3.7. Total ore studiu individual	80				
3.8. Total ore pe semestru	150				
3.9. Numărul de credite	6				

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Arhitectura Sistemelor de Calcul, Programarea Procedurală, Programarea Procedurală Avansată
4.2. de competențe	Cunoașterea programării procedurale (paradigme și limbaj, fundamente și elemente avansate) Cunoașterea principalelor structuri de date și a algoritmilor de prelucrare a acestora

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	sală de curs multimedia cu videoproiector și conexiune la Internet
5.2. de desfășurare a seminarului/laboratorului	sală de laborator echipată cu rețea de calculatoare și software corespunzător pentru dezvoltarea de programe în limbajul C

¹⁾ Adaptare după Ordinul Ministrului Educației, Cercetării, Tineretului și Sportului nr. 5703/2011 privind implementarea Codului Național al Calificărilor din învățământul superior, publicat în Monitorul Oficial al României, partea I, nr.880 bis / 13.XII.2011

6. Competen e specifice acumulate

Competen e profesionale	<ul style="list-style-type: none"> • C1.3 Elaborarea codurilor surs adecvate i testarea unitar a unor componente într-un limbaj de programare cunoscut, pe baza unor specifica ii de proiectare date. • C1.4 Testarea unor aplica ii pe baza unor planuri de test. • C1.5 Dezvoltarea de unit i de program și elaborarea documenta iilor aferente. • C2.1 Identificarea de metodologii adecvate de dezvoltare a sistemelor software • C2.3 Utilizarea metodologiilor, mecanismelor de specificare i a mediilor de dezvoltare pentru realizarea aplica iilor informatice • C3.1 Descrierea de concepte, teorii si modele folosite in domeniul de aplicare. • C3.2 Identificarea si explicarea modelelor informatice de baza adecvate domeniului de aplicare. • C3.3 Utilizarea modelelor si instrumentelor informatice si matematice pentru rezolvarea problemelor specifice domeniului de aplicare. • C3.4 Analiza datelor si a modelelor. • C4.2 Interpretarea de modele matematice i informatice (formale). • C4.3 Identificarea modelelor si metodelor adecvate pentru rezolvarea unor probleme reale. • C4.5 Incorporarea de modele formale în aplica ii specifice din diverse domenii.
Competen e transversale	<ul style="list-style-type: none"> • CT1. Aplicarea regulilor de munc organizat i eficient , a unor atitudini responsabile fa de domeniul didactic, tiin ific i profesional, în vederea valorific rii creative a propriului poten ial, cu respectarea principiilor i normelor de etic profesional ; • CT2. Desf urarea eficient a activit ilor organizate în echip i dezvoltarea capacit ilor empatic e i de comunicare inter-personal , de rela ionare i colaborare cu persoane i grupuri diverse implicate în dezvoltarea i utilizarea de sisteme software; • CT3. Utilizarea de metode i tehnici eficiente de înv are, informare, cercetare i dezvoltare a capacit ilor de valorificare a cuno tin elor, dar i de adaptare la cerin ele unei societ i dinamice i în continu schimbare, precum i dezvoltarea capacit ii de a comunica eficient i profesionist atît în limba român , cît i într-o limb de circula ie interna ional , prin însu irea i folosirea adecvat a limbajului de specialitate.

7. Obiectivele disciplinei (reie ind din grila competen elor specifice acumulate)

7.1. Obiectivul general al disciplinei	<p><i>Dobîndirea de c tre studen i a cuno tin elor fundamentale despre structurile de date elementare i algoritmii folosi i în dezvoltarea de software de bun calitate, cuno tin e referitoare la în alegerea nevoii de structuri de date, la abstractizare i tipuri de date abstracte, precum i la implementarea acestora ca structuri de date (liste, stive, cozi, arbori, grafuri etc.) i la alegerea i folosirea lor, al turi de algoritmii cei mai potrivi i, pentru dezvoltarea de aplica ii informatice complexe care func ioneaz eficient.</i></p>
7.2. Obiectivele specifice	<p><i>Dup parcurgerea disciplinei studen ii vor putea s :</i></p> <ul style="list-style-type: none"> • Explice de ce sînt necesare tipurile de date abstracte, respectiv structurile de date, i care este leg tura dintre ele; • Descrie par ial corect i s implementeze corespunz tor (în limbajul C) tipurile de date abstracte fundamentale (liste, stive, cozi, arbori, grafuri), dar i s descrie tipuri noi pe baza acestora; • Compare principalele implement ri ale tipurilor de date abstracte fundamentale ca structuri de date, dup complexitatea opera ilor elementare asupra acestora, i, în consecin , s aleag cele mai potrivite structuri de date i cei mai potrivi i algoritmi pentru a dezvolta eficient un anumit program sau pachet de programe; • Explice importan a alegerii celor mai potrivite structuri de date, respectiv algoritmi, pentru rezolvarea unei probleme date, care poate fi „solu ionat ” cu ajutorul unui program de calculator, dar i s evalueze critic diversele posibilit i i s o aleag pe cea mai bun într-un anumit context.

8. Conținuturi

8.1. Curs	Nr. ore	Metode de predare	Observații
1. Introducere - De ce avem nevoie de structuri de date. Tipuri de date abstracte elementare. Etapele elaborării unui program pentru rezolvarea unei probleme „computerizabile”.	2	<ul style="list-style-type: none"> prelegeri active și angajante; 	
2. Tipuri de date abstracte (TDA) - Conceptul de TDA. Descrierea axiomatice a TDA. Metodă de construcție a unei prezentații de TDA parțial corect. Definiții axiomatice parțial corecte pentru TDA elementare	4	<ul style="list-style-type: none"> supervizare și mentorat “deschise”; 	
3. Liste - Concepte de bază. TDA listă. Implementări posibile (vector, înlocuitor: simplă, cu header). Complexități. Funcții elementare de lucru cu liste. Liste sortate. Liste dublu înlocuitor. Liste de liste. Aplicații complexe cu liste circulare	4	<ul style="list-style-type: none"> învățarea prin descoperire; învățare pe grupuri; 	
4. Stive și cozi - Concepte de bază pentru stive. TDA stivă. Implementări posibile (vector, înlocuitor). Complexități. Funcții elementare de lucru cu stive. Aplicații complexe cu stive: evaluarea expresiilor aritmetice Concepte de bază pentru cozi. TDA coadă. Implementări posibile (vector circular, listă cu header). Complexități. Funcții elementare de lucru cu cozi. Aplicații complexe cu cozi: sortarea după ranguri	6	<ul style="list-style-type: none"> învățare bazată pe proiecte și studii de caz; învățare bazată pe rezolvarea de probleme; învățare centrată pe student; 	
5. Arbori - Concepte de bază. TDA arbore. Implementări posibile (vector tăiat, cu pointer-i). Complexități. Funcții elementare de lucru cu arbori. Aplicații complexe cu arbori multiplii	6	<ul style="list-style-type: none"> learning by doing; brainstorming; 	
6. Grafuri - Concepte de bază. TDA graf. Implementări posibile (matrice sau liste de adiacență). Complexități. Funcții elementare de lucru cu grafuri. Aplicații	2	<ul style="list-style-type: none"> învățare hibridă cu folosirea resurselor 	
7. Alte tipuri de date abstracte. Tabele de dispersie. Dicționare. Heap-uri. Arbori: echilibrați, AVL, multiplii, bicolori, TRIE.	2	<ul style="list-style-type: none"> educațională open; 	
8. Alegerea celei mai potrivite structuri de date și a celui mai potrivit algoritmului. Complexități. Comparări.	2	<ul style="list-style-type: none"> învățare reflectivă etc. 	
Bibliografie <ol style="list-style-type: none"> Carrano F. M., Savitch W., <i>Data Structures and Abstractions with Java</i>, Prentice Hall 2003 Cristea V., Athanasiu I., Kalisz E., Iorga V., <i>Tehnici de programare</i>, Teora, 1993 Knuth D.E., <i>The Art of Computer Programming Vol3: Sorting and searching</i>, Addison Wesley, Reading, Mass., 1968, Teora, 2000 Knuth D.E., <i>The Art of Computer Programming. Vol1: Fundamental Algorithms</i>, Addison Wesley, Reading, Mass., 1968, Teora, 1999 Livovschi L., Georgescu H., <i>Bazele informaticii. Algoritmi. Elaborare și complexitate</i>, Univ. din București, Fac. de Matematică, București, 1985. Livovschi L., Georgescu H. <i>Sinteza și analiza algoritmilor</i>, Universitatea din București, Fac. de Matematică, București, 1985 McConnell, S., <i>Code Complete: A practical handbook of software construction</i>, Microsoft Press, 2004* Skienna, S. S., <i>The algorithm design manual</i>, Springer-Verlag, 2010* Tomescu I., <i>Structuri de Date</i>, Editura Universității București, 1997 Tomescu I., <i>Ce este Teoria Grafurilor?</i> Ed. științifică și Enciclopedică, București, 1982 Tremblay J. P., Cheston, G. A., <i>Data Structures and Software Development in an Object Oriented Domain</i>, Prentice Hall 2003 Waite M., Lafore R., <i>Structuri de date și algoritmi în Java</i>, Teora, 2000 Wirth, N., <i>Algorithms + Data Structures = Programs</i>, Prentice Hall, 1976* Wirth, N., <i>Algorithms and Data Structures</i>, Prentice Hall, 1985* Resurse educaționale disponibile la http://www.unde.ro/cursuri/ASD/ și http://www.unde.ro/cursuri/OCW/ * Disponibile la biblioteca departamentului ITIMF 			

8.2. Seminar / laborator/proiect	Nr. ore	Metode de predare	Observații
Implementarea TDA elementare (vector, mulțime ordonată)	2	<ul style="list-style-type: none"> învățarea prin descoperire; învățarea pe grupuri; învățarea bazată pe rezolvarea de probleme; învățarea centrată pe student; learning by doing; brainstorming; învățarea hibridă; folosirea resurselor educaționale open; învățarea reflectivă etc. 	
Liste. Liste sortate. Liste dublu în lănuite. Liste de liste.	8		
Stive și cozi. Reprezentare, parcurgere și alte prelucrări elementare.	8		
Arbori. Reprezentare, parcurgere și alte prelucrări elementare.	10		
Grafuri. Reprezentare, parcurgere și alte prelucrări elementare.	10		
Alte tipuri de date abstracte. Reprezentare, parcurgere și alte prelucrări elementare.	4		
Bibliografie <ol style="list-style-type: none"> McConnell, S., <i>Code Complete: A practical handbook of software construction</i>, Microsoft Press, 2004* Waite M., Lafore R., <i>Structuri de date și algoritmi în Java</i>, Teora, 2000 Resurse educaționale disponibile la http://www.unde.ro/cursuri/ASD/ și http://www.unde.ro/cursuri/OCW/ <p>* Disponibile la biblioteca departamentului ITIMF</p>			

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

- Conținuturile disciplinei corespund cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului, așa cum rezultă din prezenta fișă, dar și din fișa de specializare, acestea fiind în concordanță deplină cu CNCIS și COR;
- Disciplina de față recomandă rile IEEE/CS și ACM legate de planul de învățământ și de conținuturile necesare pentru specializarea Informatică / Ingineria Calculatoarelor;
- Disciplina de față există în planurile de învățământ al tuturor marilor universități din România și din străinătate.

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Dobândirea competențelor profesionale și transversale specifice disciplinei	<i>Evaluare prin probă practică</i> : dezvoltarea de programe C pentru manipularea structurilor de date fundamentale, pe baza algoritmilor specifici	Evaluare program C 45%
10.5. Seminar/ laborator/proiect		<i>Evaluare orală</i> : modificarea programului inițial pentru a corespunde unui set de cerințe diferite, analiza de complexitate, eficientizarea programului inițial prin modificarea structurilor de date și/sau a algoritmilor folosiți etc.	Examinare orală 45% Din oficiu 10%
10.6. Standard minim de performanță			
<ul style="list-style-type: none"> RNCIS: Implementarea și documentarea de unități de program în limbaje de programare de nivel înalt și folosirea eficientă a mediilor de programare; Modelarea și rezolvarea unor probleme cu grad redus de complexitate, folosind cunoștințe de matematică și informatică; Realizarea și întreținerea unor aplicații informatice pentru rezolvarea unor probleme reale de complexitate redusă; Realizarea componentelor informatice pentru o aplicație dedicată de complexitate medie; 			

- *Dezvoltarea de programe C cu un grad redus de dificultate pentru manipularea unor structuri de date fundamentale (liste, stive, cozi), implementate dinamic, pe baza algoritmilor specifici.*

Data completării

26.09.2016

Semnătura titularului de curs

Semnătura titularului de seminar/laborator

Data avizării în departament

29.09.2016

Semnătura directorului de departament