

# 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 Științe
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	Metode avansate de programare
2.2. Titularul activităților de curs	Lector dr. Daniela Tudorică
2.3. Titularul activităților seminar/laborator	Lector dr. Daniela Tudorică
2.4. Titularul activității proiect	-
2.5. Anul de studiu	I
2.6. Semestrul*	2
2.7. Tipul de evaluare	Examen
2.8. 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. laborator	2	3.4. Proiect	-
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. laborator	28	3.8. Proiect	-
3.9. 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							30
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							20
Tutoriat							
Examinări							14
Alte activități							-
3.10 Total ore studiu individual	94						
3.11. Total ore pe semestru	150						
3.12. Numărul de credite	6						

## 4. Precondiții

4.1. de curriculum	Fundamentele programării, Algoritmi fundamentali
4.2. de competențe	Proiectarea algoritmilor, cunoștințe de programare în limbajul C

## 5. Condiții

5.1. de desfășurare a cursului	<ul style="list-style-type: none"><li>➤ sală de curs multimedia pentru realizarea de prelegeri, conversații, dezbateri</li><li>➤ suport electronic: <a href="http://www.ls.upg-elearning.ro">www.ls.upg-elearning.ro</a> <a href="https://timf.upg-ploiesti.ro/cursuri/MAP">https://timf.upg-ploiesti.ro/cursuri/MAP</a></li></ul>
--------------------------------	--

<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

5.2. de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>➤ laborator dotat cu sisteme de calcul cu mediu integrat de dezvoltare</li> <li>➤ instalat (CodeBlocks, Visual Studio) pentru scrierea de programe în limbajul C++</li> </ul>
---	--

## 6. Competențe specifice acumulate

<b>Competențe profesionale</b>	<p>C1.2 EXPLICAREA UNOR APLICAȚII SOFT EXISTENTE, PE NIVELURI DE ABSTRACTIZARE (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>
<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> <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 metodelor avansate de programare în limbajul C++.
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:</p> <ul style="list-style-type: none"> <li>➤ să elaboreze algoritmi de rezolvare a unor probleme practice;</li> <li>➤ să codifice algoritmi în limbajul C++, utilizând facilitățile noilor standarde (C++17, C++20);</li> <li>➤ să utilizeze un mediu integrat de dezvoltare (CodeBlocks, Visual Studio) pentru scrierea de programe în limbajul C++;</li> <li>➤ să 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.</li> </ul>

## 8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
1. Recapitulare elemente de programare din limbajul C.	2		
2. Elemente specifice ale limbajului C++: generalități,	7		

structura unui program C++, biblioteci, spații de nume, instrucțiuni I/O, tipuri de date; operatori, tablouri, structuri, pointeri, referințe. Stringuri în C++. Fișiere în C++. Directive preprocesor în C++. Tratarea excepțiilor în C++		Expunere pe baza suportului de curs, studii de caz, conversație, dezbateri	
3. Funcții în C++ (definirea, declararea, apelul unei funcții, transmiterea parametrilor unei funcții, supraîncărcarea funcțiilor, funcții inline). 4. Funcții recursive. Strategii iterative vs recursive	3		
5. Funcții template. Clase template. Iteratori. Containere – în C++. Biblioteci C++ (biblioteca algorithm). 6. Programare vizuală, interfețe grafice în C++.	4		
7. Analiza complexității algoritmilor (tipuri de analiză a complexității, calculul complexității algoritmilor). Complexitate computațională avansată. Complexitate exactă, complexitate aproximativă. Notăția asimptotică – O, o, Ω, Θ. Eficiența timp și spațiu. Clase de complexitate a problemelor. Clase de probleme: P și NP, NP-complete, NP-dificile. Complexitatea timp a funcțiilor recursive (Teorema master)	4		
8. Strategii algoritmice și analize de complexitate: • Brute-force • Metode de reducere a dimensiunii problemei: transform and conquer • Divide et impera • Backtracking • Algoritmi de sortare	6		
9. Recapitulare. Discutarea subiectelor de examen.	2		
<b>Bibliografie</b> <ol style="list-style-type: none"> <li>1. McConnell, Jeffrey J., Analysis of algorithms: An active learning approach, Jones and Bartlett Publishers, 2008</li> <li>2. Coșulschi M., Algoritmi fundamentali. Proiectare și implementare, Ed. Universitaria, Craiova, 2015</li> <li>3. Levitin A., Introduction to the design and analysis of algorithms, Pearson Education, third edition, 2012</li> <li>4. Marinoiu C., Programarea în limbajul C, Editura Universității din Ploiești, 2000</li> <li>5. Stroustrup B., The C++ Programming Language, Addison-Wesley, 2018</li> <li>6. Stroustrup B., Programming: Principles and Practice Using C++, Second Edition, Addison-Wesley, 2014</li> <li>7. Schildt Herbert, The complete reference C++, Mc Graw Hill Higher Education, 2014</li> <li>8. Weiss, Mark Allen, Data structures and algorithm analysis in C++, Pearson Addison Wesley, 2006</li> <li>9. Stevanovic, Milan - <i>Advanced C and C++ Compiling</i> - Apress, 2014, disponibilă la biblioteca ITIMF</li> </ol>			
<b>8.2. Laborator</b>	Nr. ore	Metode de predare	Observații
1. Recapitulare – Fundamentele programării (limbajul C):	2		

declararea și inițializarea variabilelor, instrucțiuni pentru codificarea operațiilor de bază și a structurilor de control dintr-un algoritm.		Expunere, studii de caz, conversație, dezbateri, aplicații pe calculator	
Programarea în limbajul C++: • Aplicații cu tablouri (vectori și matrice) și pointeri. • Aplicații cu stringuri (clasa string). • Funcții definite de utilizator. Supraîncărcarea funcțiilor. Funcții recursive. • Aplicații cu structuri. Aplicații cu fișiere. • Directive preprocesor. Tratarea excepțiilor. • Definirea și utilizarea funcțiilor template. Lucrul cu iteratori și containere: array, vector. • Interfețe grafice în C++.	2 2 2 2 2 2 2		
Strategii algoritmice și analize de complexitate: • Brute-force • Metode de reducere a dimensiunii problemei: transform and conquer • Divide et impera • Backtracking (iterativ și recursiv) • Algoritmi de sortare	10		
5.Examinare orală cu bilete: aplicație pe calculator	2		

#### Bibliografie

1. Manual online de C++ (CPlusPlus): <http://www.cplusplus.com/doc/tutorial/>
2. Curs online de C++ (LearnCpp): <http://www.learncpp.com/>
3. Curs online de C++ (TutorialsPoint): <http://www.tutorialspoint.com/cplusplus/index.htm>
4. McConnell, Jeffrey J., Analysis of algorithms: An active learning approach, Jones and Bartlett Publishers, 2008
5. Alexandrescu A., Modern C++ Design, Addison-Wesley, 2001
6. Coșulschi M., Algoritmi fundamentali. Proiectare și implementare, Ed. Universitaria, Craiova, 2015
7. Levitin A., Introduction to the design and analysis of algorithms, Pearson Education, third edition, 2012
8. Marinoiu C., Programarea în limbajul C, Editura Universității din Ploiești, 2000
9. Stroustrup B., The C++ Programming Language, Addison-Wesley, 2018
10. Stroustrup B., Programming: Principles and Practice Using C++, Second Edition, Addison-Wesley, 2014
11. Schildt Herbert, The complete reference C++, Mc Graw Hill Higher Education, 2014
12. Thomas H. Cormen, et al., Introduction to Algorithms, Third Edition, The MIT Press, 2009
13. Weiss, Mark Allen, Data structures and algorithm analysis in C++, Pearson Addison Wesley, 2006

### **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 de proiectare și implementare a aplicațiilor C++, cunoștințe indispensabile în cazul în care absolventul decide să urmeze o carieră în domeniul

software

## 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, capacitatea de a opera cu cunoștințele asimilate, capacitatea de analiză, de interpretare personală, gradul de asimilare a limbajului de specialitate și capacitatea de comunicare.	Test grilă	40%
10.5. Seminar/laborator	Aplicarea practică a elementelor prezentate la orele de curs și de laborator, prin realizarea de aplicații în limbajul C++.	Teste de evaluare pe parcursul semestrului Intrebari privitoare la modul de rezolvare a problemelor propuse in cadrul laboratorului	60%
10.6. Standard minim de performanță			
<ul style="list-style-type: none"><li>➤ Asimilarea limbajului de specialitate privind programarea avansată în C++.</li><li>➤ Realizarea unor programe C++ pentru rezolvarea unor probleme de dificultate medie</li></ul>			

Data completării

22 sept. 2024

Semnătura titularului de

curs

Lector dr. Daniela Tudorică

Semnătura titularului de laborator

Lector dr. Daniela Tudorică

Data avizării în  
departament

24 sept. 2024

Director de departament  
Lector dr. Anca Baci

Decan  
Prof. univ. dr. Mihaela Suditu