

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	Metode avansate de programare
2.2. Titularul activităților de curs	Lector dr. Șchiopu Daniela
2.3. Titularul activităților seminar/laborator	Lector dr. Aldescu Iuliana
2.4. Titularul activității proiect	-
2.5. Anul de studiu	1
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. Seminar/laborator	2	3.4. Proiect	-
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. Seminar/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							0
3.10 Total ore studiu individual	94						
3.11. Total ore pe semestru	150						
3.12. Numărul de credite	6						

4. Precondiții (acolo unde este cazul)

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

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	➤ sală de curs multimedia pentru realizarea de prelegeri, conversații, dezbateri
5.2. de desfășurare a seminarului/laboratorului	➤ laborator dotat cu sisteme de calcul cu mediu integrat de dezvoltare instalat (Code Blocks) pentru scrierea de programe în limbajul C++

6. Competențe specifice acumulate

Competențe profesionale	<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>
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 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">➤ să elaboreze algoritmi de rezolvare a unor probleme practice;➤ să codifice algoritmi în limbajul C++, utilizând facilitățile noilor standarde (C++14, C++17);➤ să utilizeze un mediu integrat de dezvoltare (Code Blocks) pentru scrierea de programe în limbajul C++;➤ 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.

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, 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	7		

excepțiilor în C++.			
3. Funcții în C++ (definirea, declararea, apelul unei funcții, transmiterea parametrilor unei funcții, supraîncărcarea funcțiilor, funcții inline).	3	Expunere, studii de caz, conversație, dezbateri, utilizare tehnologii multimedia	
4. Funcții recursive. Strategii iterative vs recursive.			
5. Funcții template. Clase template. Iteratori. Containere – în C++.	4		
6. Programare vizuală, interfețe grafice în C++.			
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: <ul style="list-style-type: none"> • 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		
Bibliografie			
1. McConnell, Jeffrey J., <i>Analysis of algorithms: An active learning approach</i> , Jones and Bartlett Publishers, 2008 2. Alexandrescu A., <i>Modern C++ Design</i> , Addison-Wesley, 2001 3. Coșulschi M., <i>Algoritmi fundamentali. Proiectare și implementare</i> , Ed. Universitaria, Craiova, 2015 4. Levitin A., <i>Introduction to the design and analysis of algorithms</i> , Pearson Education, third edition, 2012 5. Marinoiu C., <i>Programarea în limbajul C</i> , Editura Universității din Ploiești, 2000 6. Stroustrup B., <i>The C++ Programming Language</i> , Addison-Wesley, 2018 7. Stroustrup B., <i>Programming: Principles and Practice Using C++</i> , Second Edition, Addison-Wesley, 2014 8. Schildt Herbert, <i>The complete reference C++</i> , Mc Graw Hill Higher Education, 2014 9. Thomas H. Cormen, et al., <i>Introduction to Algorithms</i> , Third Edition, The MIT Press, 2009 10. Weiss, Mark Allen, <i>Data structures and algorithm analysis in C++</i> , Pearson Addison Wesley, 2006			
8.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
1. Recapitulare – Fundamentele programării (limbajul C): declararea și inițializarea variabilelor, instrucțiuni pentru codificarea operațiilor de bază și a structurilor de control dintr-un algoritm.	2	Dezbateri, studii de caz, exemple, angajarea fiecărui student în descoperirea cunoștințelor.	
2. Programarea în limbajul C++:			
<ul style="list-style-type: none"> • Aplicații cu tablouri (vectori și matrice) și pointeri. • Aplicații cu stringuri (clasa <i>string</i>). • Funcții definite de utilizator. Supraîncărcarea funcțiilor. 	2		
Funcții recursive.	2		

<ul style="list-style-type: none"> • Aplicații cu structuri. Aplicații cu fișiere. • Directive preprocesor. Tratarea excepțiilor. • Definirea și utilizarea funcțiilor template. Lucrul cu iteratori și containere: <i>array</i>, <i>vector</i>. • Interfețe grafice în C++. 	2		
<ul style="list-style-type: none"> • Brute-force • Metode de reducere a dimensiunii problemei: transform and conquer • Divide et impera • Backtracking (iterativ și recursiv) • Algoritmi de sortare 	10		
4. Evaluare laborator	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. Marinou 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.	Examen scris	50% Pentru promovarea examenului, minim nota 5 la proba scrisă.
10.5. Seminar/laborator	Aplicarea practică a elementelor prezentate la orele de curs și de laborator, prin realizarea de aplicații în limbajul C++.	Evaluare orală	50% Pentru promovarea examenului, minim nota

			5 la laborator.
10.7. Standard minim de performanță			
<ul style="list-style-type: none"> ➤ Asimilarea limbajului de specialitate privind programarea avansată în C++. ➤ Realizarea unor programe C++ pentru rezolvarea unor probleme de dificultate medie. 			

Data completării
14.09.2021

Semnătura titularului de curs
Lect. dr. Daniela Șchiopu

Semnătura titularului de laborator
Lector dr. Aldescu Iuliana

Data avizării în departament
16.09.2021

Director de departament
Conf. dr. Gabriela Moise

Decan
Prof. univ. dr. Mihaela Suditu