

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Master
1.6 Programul de studiu / Calificarea	Informatică didactică

2. Date despre disciplină

2.1 Denumirea disciplinei	Metode și tehnici de programare (pentru perfecționarea profesorilor)						
2.2 Titularul activităților de curs	Asist. dr. BOCICOR Maria Iuliana						
2.3 Titularul activităților de seminar	Asist. dr. BOCICOR Maria Iuliana						
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2	3.3 seminar/laborator	1
3.4 Total ore din planul de învățământ	42	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					36
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					44
Tutoriat					18
Examinări					10
Alte activități:					
3.7 Total ore studiu individual	133				
3.8 Total ore pe semestru	175				
3.9 Numărul de credite	7				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	
4.2 de competențe	Abilități medii de programare

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> • Sală de curs cu videoproiector
5.2 De desfășurare a seminarului/laboratorului	

6. Competențele specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> • Aprofundarea cunoștințelor legate de metodele de proiectare a algoritmilor și de programare. • Aprofundarea cunoștințelor legate de diverse tehnici de programare. • Aprofundarea cunoștințelor legate de noțiunile și conceptele de bază ale limbajului C standard. • Formarea deprinderilor și identificarea metodelor și tehnicilor adecvate pentru rezolvarea unor probleme reale • Capacitate de analiză și sinteză.
Competențe transversale	<ul style="list-style-type: none"> • Aplicarea regulilor de muncă organizată și eficientă, cu respectarea principiilor și a normelor de etică profesională. • Manifestarea unor atitudini responsabile față de domeniul didactic și științific. • Capacitate de muncă independentă. • Utilizarea unor metode și tehnici eficiente de învățare și informare și dezvoltarea capacităților de valorificare a cunoștințelor.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Studiarea unor metode și tehnici de programare pentru rezolvarea unor probleme reale, folosind limbajul C standard.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Studiarea noțiunilor și conceptelor de bază ale limbajului C standard. • Formarea deprinderilor de utilizare a unor metode și tehnici de proiectare a algoritmilor și de programare. • Identificarea metodelor și tehnicilor adecvate de programare și utilizarea limbajului C standard pentru rezolvarea unor probleme reale • Însușirea unui stil de programare conform celor mai bune recomandări practice.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Limbajul de programare C. Introducere <ul style="list-style-type: none"> • Limbaje de programare high-level și low-level • Structura unui program C • Variabile și constante • Tipuri de date și operatori în C • Conversia tipurilor: conversii implicite/explicite • Citirea și scrierea datelor în C 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
2 – 3. Metode de proiectare și programare <ul style="list-style-type: none"> • Metode de proiectare a algoritmilor: <ul style="list-style-type: none"> • Proiectarea ascendentă • Proiectarea descendentă • Proiectarea modulară 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	

<ul style="list-style-type: none"> • Module în C • Programarea structurată. Structuri de bază • Structurile <i>if-then-else</i>, <i>switch-case</i> • Structurile repetitive <i>for</i>, <i>while</i>. Structuri repetitive infinite. 		
<p>4. Preprocesorul C</p> <ul style="list-style-type: none"> • Rolul preprocesorului • Directivele <i>#include</i>, <i>#define</i>, <i>#undef</i> • Compilarea condițională: directivele <i>#if</i>, <i>#ifdef</i>, <i>#ifndef</i>, <i>#endif</i> • Include guards • Macro-operatori • Evitarea compilării multiple 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>5. Tipuri structurate de date</p> <ul style="list-style-type: none"> • Tablouri unidimensionale și bidimensionale • Șiruri de caractere și operații pe acestea • Definierea și utilizarea înregistrărilor (<i>struct</i>) în C • Înregistrări încuibate • Tipurile de date <i>union</i>, <i>enum</i> • Definierea unor structuri de date pentru operarea cu liste, stive, cozi. 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>6. Subalgoritmi și subprograme</p> <ul style="list-style-type: none"> • Noțiunea de subprogram • Variabile globale și locale, vizibilitatea variabilelor • Parametri formali și parametri actuali • Transmiterea parametrilor • Funcții inline • Funcții recursive 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>7 – 9. Alocarea dinamică a memoriei</p> <ul style="list-style-type: none"> • Pointeri și aritmetica pointerilor • Adrese, referințe, dereferențiere, operatorul <i>sizeof</i> în C • Alocarea și dealocarea memoriei: operatorii <i>malloc</i> (<i>realloc</i>, <i>calloc</i>)/<i>free</i> în C • Utilizarea pointerilor: avantaje și dezavantaje • Argumentele liniei de comandă în C • Echivalența între tablouri unidimensionale și pointeri • Tablouri bidimensionale reprezentare cu ajutorul pointerilor • Pointeri și constante • Tipul void și pointeri la void • Înregistrări conținând pointeri • Tablouri conținând înregistrări/pointeri la înregistrări • Pointeri la funcții 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
<p>10. Tehnici de programare – Divide et impera</p> <ul style="list-style-type: none"> • Prezentare – considerații teoretice • Aplicații clasice și aplicații mai complexe 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	

11. Tehnici de programare – Backtracking I <ul style="list-style-type: none"> • Prezentare – considerații teoretice • Aplicații clasice și aplicații mai complexe 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
12. Tehnici de programare – Backtracking II <ul style="list-style-type: none"> • Aplicații complexe în plan (bidimensional) 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
13. Tehnici de programare – Greedy <ul style="list-style-type: none"> • Prezentare – considerații teoretice • Aplicații clasice și aplicații mai complexe 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	
14. Tehnici de programare – Programarea dinamică <ul style="list-style-type: none"> • Prezentare – considerații teoretice • Aplicații clasice și aplicații mai complexe 	<ul style="list-style-type: none"> • Expunerea interactivă • Explicația • Conversația • Demonstrația didactică 	

Bibliografie

1. Cormen T.H., Leiserson C.E, Rivest R.L, Stein C. – Introduction to Algorithms (third edition), ISBN: 9780262533058, 2001.
2. Knuth D.E. – The Art of Computer Programming, Addison-Wesley, 1968.
3. Levitin A. – Introduction to the Design & Analysis of Algorithms (third edition), Addison-Wesley, 2002.
4. Kernighan B.W., Ritchie D.M. – The C Programming Language, 2nd Edition, Englewood Cliffs, NJ: Prentice Hall 1988, ISBN 0-13-110362-8.
5. Kalicharan N. C By Example. Cambridge University Press, 1996.
6. Frențiu M., Pop H.F., Șerban G. – Programming Fundamentals, Ed.Presa Universitara Clujeana, Cluj-Napoca, 2006.
7. Boian F., Vancea A. Iurian S. Limbajul C, culegere de probleme. Lito UBB, 1992.

8.2 Seminar	Metode de predare	Observații
1. Aplicații practice simple <ul style="list-style-type: none"> • Rezolvarea unor probleme folosind programarea modulară în limbajul C. 	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Exercițiul 	
2. Tablouri și tipuri structurate de date <ul style="list-style-type: none"> • Implementarea în C a unor algoritmi fundamentali de prelucrare a datelor structurate în tablouri (căutare secvențială și binară, sortări, interclasare). • Programare prin abstractizarea datelor. Folosirea înregistrărilor în C. 	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Exercițiul 	
3. Alocarea dinamică a memoriei <ul style="list-style-type: none"> • Rezolvarea unor probleme simple utilizând alocarea dinamică a memoriei în C. • Abordarea unor probleme reale pornind de la utilizarea tipurilor abstracte de date și definirea structurilor de date necesare, folosind pointeri, în C. 	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Exercițiul 	
4. Biblioteca IO standard în C. Lucrul cu fișiere <ul style="list-style-type: none"> • Citirea și scrierea datelor în fișiere text • Structura <i>FILE</i> • Utilizare funcțiilor din biblioteca IO standard pentru 	<ul style="list-style-type: none"> • Explicația • Conversația • Modelarea • Exercițiul 	

lucrul cu fișiere.		
5. Stabilirea problemei pentru proiectul software – problemă practică reală, a cărei rezolvare presupune utilizarea uneia dintre tehnicile de programare prezentate la curs.	<ul style="list-style-type: none"> • Conversația • Dezbateră • Studiul de caz 	
6. Prezentare de referate și rapoarte privind derularea proiectelor	<ul style="list-style-type: none"> • Conversația • Dezbateră • Studiul de caz • Prezentări 	
7. Prezentarea aplicației realizate pentru proiectul software	<ul style="list-style-type: none"> • Prezentări • Demo-uri 	
Bibliografie Studenții vor: <ul style="list-style-type: none"> • folosi documentația menționată în cadrul bibliografiei de la curs • căuta noi surse de informare referitoare la metode și tehnici de programare și la limbajul C • căuta articole de cercetare în legătură cu aplicațiile tehnicilor de programare prezentate în principalele baze de date cu lucrări de informatică sau domenii adiacente. 		

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

<ul style="list-style-type: none"> • Conținutul disciplinei este în concordanță cu ceea ce se face în alte centre universitare din țară și din străinătate. • Conținutul disciplinei asigură cunoștințele fundamentale necesare pentru rezolvarea de probleme folosind metodele și tehnicile consacrate de programare, precum și cunoștințele fundamentale pentru dezvoltarea de aplicații folosind limbajul C standard.
--

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală(%)
10.4 Curs	<ul style="list-style-type: none"> • Cunoașterea conceptelor de bază legate de metodele și tehnicile de programare prezentate. • Cunoașterea elementelor de bază ale limbajului C standard și capacitatea de a le utiliza în vederea dezvoltării de aplicații software. 	Examen scris	40%
10.5 Seminar/ activități laborator	<ul style="list-style-type: none"> • Capacitatea de a aplica cunoștințele teoretice dobândite pentru dezvoltarea unor aplicații software în C. • Capacitatea de a studia literatura de specialitate în domeniul tematicii cursului. • Capacitatea de a dezvolta 	<ul style="list-style-type: none"> • Referat teoretic • Proiect software • Prezența la seminar/laborator • Oficiu 	20% 20% 10% 10%

	un proiect software.		
10.6 Standard minim de performanță			
• Cel puțin nota 5 (pe o scară de la 1 la 10) la examenul scris, referat și proiect.			

Data completării

20.04.2016

Semnătura titularului de curs

Asist. dr. Bocicor Maria Iuliana

Semnătura titularului de seminar

Asist. dr. Bocicor Maria Iuliana

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Andreica Anca