

FIȘA DISCIPLINEI

Anul universitar 2025 - 2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Lucian Blaga din Sibiu
1.2. Facultatea	Științe
1.3. Departament	Matematică și Informatică
1.4. Domeniul de studiu	Informatică
1.5. Ciclul de studii ¹	Licență
1.6. Specializarea	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	Programare orientată obiect				Cod	FSTI.MAI.INF.L.SO.2 .2020.E-5.1
2.2. Titular activități de curs	Lector univ. dr. Mircea-Adrian Mușan					
2.3. Titular activități practice	Lector univ. dr. Costel Maniu					
2.4. An de studiu ²	1	2.5. Semestrul ³	2	2.6. Tipul de evaluare ⁴	E	
2.7. Regimul disciplinei ⁵	O		2.8. Categoria formativă a disciplinei ⁶			DF

3. Timpul total estimat

3.1. Extinderea disciplinei în planul de învățământ – <i>număr de ore pe săptămână</i>					
3.1.a.Curs	3.1.b. Seminar	3.1.c. Laborator	3.1.d. Proiect	3.1.e Alte	Total
2		2			4
3.2. Extinderea disciplinei în planul de învățământ – <i>total ore din planul de învățământ</i>					
3.2.a.Curs	3.2.b. Seminar	3.2.c. Laborator	3.2.d. Proiect	3.2.e Alte	Total ⁷
28		28			56
Distribuția fondului de timp pentru studiu individual⁸					Nr. ore
Studiul după manual, suport de curs, bibliografie și notițe					24
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					5
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					23
Tutoriat ⁹					14
Examinări ¹⁰					3
3.3. Total ore alocate studiului individual¹¹ (NOSI_{sem})					69
3.4. Total ore din Planul de învățământ (NOAD_{sem})					56
3.5. Total ore pe semestru¹² (NOAD_{sem} + NOSI_{sem})					125
3.6. Nr ore / ECTS					25
3.7. Număr de credite¹³					5

4. Precondiții (acolo unde este cazul)

4.1. Discipline necesar a fi promovate anterior (de curriculum) ¹⁴	Fundamentele programării / Programare procedurală
4.2. Competențe	Cunoașterea și utilizarea adecvată a noțiunilor specifice tehnicii OOP. Capacitatea de a interpreta și înțelege programe și aplicații realizate, pe baza codului sursă. Capacitatea de a proiecta și realiza programe folosind tehnica OOP, pe baza specificațiilor.

5. Condiții (acolo unde este cazul)

5.1. De desfășurare a cursului ¹⁵	- Acces platformă G Suite for Education (Google Classroom, Google Meet); - Sală de curs dotată cu tablă și videoproiector; - Nu se acceptă comportamentul agresiv, insultarea persoanelor și alte forme de agresiuni verbale sau fizice; - Cursul se desfășoară în condiții de ascultare și participare activă la discuții.
5.2. De desfășurare a activităților practice (lab/sem/pr/aplic) ¹⁶	- Acces platforma G Suite for Education (Google Classroom, Google Meet); - Laborator dotat cu tablă, videoproiector și software specializat în vederea realizării aplicațiilor.

6. Rezultate ale învățării¹⁷

Numărul de credite alocat disciplinei: 5				
Rezultatele învățării				Repartizare credite pe rezultatele învățării
Nr. crt.	Cunoștințe	Aptitudini	Responsabilitate și autonomie	
RÎ 1	Studentul/absolventul explică principiile programării orientate pe obiecte: clase, obiecte, tipuri abstracte de date și încapsulare.	Studentul/absolventul declară și implementează clase simple, definind membri și metode de bază.	Studentul/absolventul respectă principiile OOP și corectitudinea codului.	1.0
RÎ 2	Studentul/absolventul descrie și utilizează constructori, destructori, membri statici și funcții constante.	Studentul/absolventul implementează și testează clase cu inițializare corectă și gestionare a ciclului de viață al obiectelor.	Studentul/absolventul își asumă responsabilitatea pentru stabilitatea și claritatea implementării.	1.0
RÎ 3	Studentul/absolventul explică și aplică mecanisme de supraîncărcare a operatorilor și utilizarea funcțiilor/claselor prietene.	Studentul/absolventul implementează supraîncărcarea operatorilor și gestionează interacțiunile între obiecte.	Studentul/absolventul verifică corectitudinea și reutilizabilitatea codului scris.	1.0
RÎ 4	Studentul/absolventul explică principiile moștenirii și polimorfismului, inclusiv rolul claselor	Studentul/absolventul proiectează ierarhii de clase, implementează polimorfism și utilizează clase abstracte.	Studentul/absolventul alege soluții potrivite pentru extensibilitate și menținere.	1.0

	abstracte și al funcțiilor virtuale.			
RÎ 5	Studentul/absolventul recunoaște metodele avansate: lucrul cu fișiere, genericitatea (șabloane, STL), tratarea excepțiilor și concepte fundamentale de proiectare.	Studentul/absolventul manipulează obiecte fișier, implementează șabloane și aplică tratamente de excepții în aplicații OOP.	Studentul/absolventul își asumă responsabilitatea pentru integritatea și robustețea aplicațiilor dezvoltate.	1.0

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1. Obiectivul general	Înșușirea cunoștințelor generale despre paradigma programării orientate pe obiecte.
7.2. Obiectivele specifice	Înșușirea unui limbaj care suportă paradigma de programare, ca instrument efectiv de lucru (limbajul C++). Elemente primare și fundamentale de proiectare. Introducerea în elementele programării vizuale prin intermediul limbajului Visual C++.

8. Conținuturi

8.1. Curs ¹⁸	Metode de predare ¹⁹	Nr. ore
Curs 1: Metoda programării orientată pe obiecte. Contextul apariției metodei. Principii ale metodei de programare. Calitatea produselor program și noua metodă. Limbajele de programare care suportă metoda de programare orientată pe obiecte. Limbajul C++, superset al limbajului C. Tipul referință. Funcții in-line. Pointeri. Tipul Struct. Pointeri Smart.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 2-3: Introducere în programarea obiectuală. Clase și obiecte. Tipul abstract de date. Încapsulare. Declararea și definirea unei clase. Declararea obiectelor și referirea membrilor. Structura spațiului obiectelor. Constructori generali. Constructori de inițializare prin copiere. Destructorii.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	4
Curs 4: Facilități avansate ale programării OPP în limbajul C++. Membri statici. Funcții membru constante. Obiecte constante. Array-uri de obiecte. Supraîncărcarea operatorilor. Funcții și clase prietene.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 5: Realizarea operațiilor de intrare și ieșire prin intermediul obiectelor. Clase și obiecte de intrare și ieșire. operații standard cu format. Controlul formatului în operații. Manipulatori. Operații fără format. Supraîncărcarea operatorilor de extracție și inserție pentru obiectele unei clase.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2

Curs 6: Supraîncărcarea operatorilor remarcabili în C++. Supraîncărcarea operatorului de atribuire. Supraîncărcarea operatorilor de relație. Supraîncărcarea operatorului de apel de funcție. Supraîncărcarea operatorului de indexare. Conversii. Supraîncărcarea operatorului de cast.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 7-9: Principiul moștenirii în OOP. Moștenirea simplă: controlul membrilor. Moștenirea multiplă: moștenirea repetată. Ierarhii de clase: constructorii și destructorii, supraîncărcarea funcțiilor, operatorul de atribuire, conversia obiectelor și a pointerilor la obiecte, redefinirea funcțiilor. Polimorfism și funcții virtuale în ierarhii de clase. Interapelarea funcțiilor în ierarhii de clase. Virtualizarea destructorilor în ierarhii de clase. Clase abstracte.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	6
Curs 10: Obiecte fișier în programarea orientată spre obiecte. Modul de lucru cu obiecte fișier. Tratarea erorilor în lucrul cu fișier. Prelucrarea fișierelor de tip text. Realizarea lucrului cu fișiere binare: accesul secvențial și accesul direct.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 11: Genericitatea în programarea orientată obiect. Funcții generice. Clase generice. Biblioteca STL de clase generice: principalele clase container. Algoritmi generici, funcții generice.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 12: Tratarea excepțiilor și portabilitatea tipurilor. Tratarea excepțiilor. Instrucțiunea throw. Lansarea excepțiilor cu o funcție din cadrul blocului try. Aplicații ale tratării excepțiilor. Restricționarea și relansarea excepțiilor. Spații de nume.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 13: Elemente primare și fundamentale de proiectare. Modelarea obiectelor: modelul obiectelor, modelul dinamic al obiectelor, modelul funcțional al obiectelor. Etapele activității de realizare a proiectului: analiza, proiectarea de ansamblu, proiectarea de detaliu.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Curs 14: Inițiere în programarea concurentă. Procese. Fire de execuție.	Expunere, prelegere, utilizare videoproiector, discuții cu studenții	2
Total ore curs:		28

8.2. Activități practice

8.2.b. Laborator		Metode de predare ²⁰	Nr. ore
Laborator 1	Utilizarea îmbunătățirilor limbajului C++ în programarea procedurală. Realizarea unor programe care utilizează referințe în transmiterea parametrilor. Realizarea de programe de lucru cu memoria dinamică utilizând operatorii new și delete. Programe care utilizează funcții cu argumente implicite și realizează supraîncărcarea funcțiilor. Realizarea programelor ce utilizează funcții in-line și operatorul de rezoluție.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 2-3	Introducere în conceptul de OOP. Aplicații ce construiesc clase, declară și alocă memorie pentru obiecte. Realizarea unor aplicații pentru exemplificarea instanțierii prin utilizarea constructorilor. Destructorii.	Expunerea, utilizare videoproiector, discuții cu studenții	4
Laborator 4	Supraîncărcarea operatorilor aritmetici. Realizarea aplicațiilor cu dezvoltare de clase ce implică funcții prieten și operatori aritmetici supraîncărcați.	Expunerea, utilizare videoproiector, discuții cu studenții	2

Laborator 5	Realizarea de aplicații prin operații de intrare/ieșire în C++. Clase de bază pentru streamuri. Utilizarea membrilor clasei IOS pentru formatarea operațiilor de I/O. Manipulatori. Supraîncărcarea operatorilor "<<" și ">>".	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 6	Realizarea de aplicații prin operatori speciali supraîncărcați în C++. Pointerul this. Supraîncărcarea operatorului "[]". Supraîncărcarea operatorului "()". Supraîncărcarea operatorului ",". Supraîncărcarea operatorului "=". Supraîncărcarea operatorilor "new" și "delete".	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 7	Construcția de aplicații folosind conceptul de moștenirea simplă în tehnica OOP. Controlul accesului la membrii moșteniți. Constructorii și destructorii în ierarhii de clase. Alte probleme generate de procesul moștenirii. Aplicații complexe utilizând procesul de moștenire simplă.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 8	Construcția de aplicații folosind moștenirea multiplă în OOP. Clase virtuale. Funcții virtuale. Polimorfism.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 9	Clase abstracte și clase prieten. Realizarea de aplicații complexe ce folosesc concepte precum: polimorfism, clase abstracte și clase prieten.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 10	Prelucrarea fișierelor text și binare în OOP. Ierarhia de clase pentru lucrul cu fișiere. Deschiderea fișierelor. Închiderea fișierelor. Intrări și ieșiri la nivel de fișier. Accesul la datele dintr-un fișier. Realizarea de aplicații prin folosirea obiectelor fișier în tehnica OOP.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 11	Funcții generice. Clase generice. Șabloane. Clase generice. Realizarea de exemple cu două clase generice.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 12	Realizarea de aplicații pentru tratarea excepțiilor. Tratarea excepțiilor. Instrucțiunea throw. Lansarea excepțiilor cu o funcție din cadrul blocului try. Aplicații ale tratării excepțiilor. Spații de nume.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 13	Introducere în proiectare. Studiu de caz.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Laborator 14	Prezentarea proiectului final realizat individual de către studenți.	Expunerea, utilizare videoproiector, discuții cu studenții	2
Total ore seminar			28

9. Bibliografie

9.1. Referințe bibliografice recomandate	<i>Object Oriented Programming, A Complete Guide</i> – 2020 Edition Paperback – April 16, 2021, ISBN-10: 1867306336, ISBN-13 : 978-1867306337
	Valer Roșca, <i>Programarea orientată pe obiecte în C++</i> , Editura Universității "Lucian Blaga", Sibiu, 2005
	Mircea Mușan, <i>Programare orientată pe obiecte</i> , note de curs și lucrări de laborator (2025) – în format electronic
9.2. Referințe bibliografice suplimentare	Ion Smeureanu, Marian Dardala, <i>Programarea orientată obiect în limbajul C++</i> , Editura: Cison, 2002, ISBN/Cod: 973-8301-06-8
	Herbert Schildt, <i>C++. Manual complet</i> , Editura Teora, București, 2000

Dr. Kris Jamsa, Lars Klander, *Totul despre C și C++. Manual fundamental de programare în C și C++*, Editura Teora, București, 2000

10. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului²¹

Se realizează prin contacte periodice cu aceștia în vederea analizei conținuturilor. Conținuturile disciplinei sunt permanent adaptate atât tradițiilor cât și evoluțiilor domeniilor în care pot fi angajați absolvenții. Acest lucru se realizează atât pe baza experienței cadrelor didactice ale departamentului în domeniul didactic și în cel IT dar și printr-o permanentă colaborare și consultare cu colegii altor universități din țară și străinătate cât și cu alți posibili angajatori din domeniul aferent programului. Astfel se insistă în formarea la studenți a unei gândiri structurate, a unui raționament organizat logico-deductiv, a capacității de analiză și sinteză, de imaginație, intuiție, de anticipare a unor rezultate. Analiza funcțională clasică este o disciplină modernă care își găsește aplicabilitate în multe și foarte variate domenii științifice.

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală	Obs. ²²
11.4a Examen / Colocviu	Cunoștințe teoretice și practice însușite (cantitatea, corectitudinea, acuratețea)	Teste pe parcurs ²³ :	$P_{1.1}=10\%$ $N_{1.1} \geq 5$	$P_1 = P_{1.1} + P_{1.2} + P_{1.3} + P_{1.4}$
		Teme de casă:	$P_{1.2} = _ \%$ $N_{1.2} \geq 5$	
		Alte activități ²⁴ :	$P_{1.3} = _ \%$ $N_{1.3} \geq 5$	
		Evaluare finală:	$P_{1.4}=50\%$ $N_{1.4} \geq 5$	
11.4b Seminar	Frecvența/relevanța intervențiilor sau răspunsurilor	Evidența intervențiilor, portofoliu de lucrări (referate, sinteze științifice)	$P_2 = _ \%$ $N_2 \geq 5$	nCPE
11.4c Laborator	Cunoașterea aparaturii, a modului de utilizare a instrumentelor specifice; evaluarea unor instrumente sau realizări, prelucrarea și interpretarea unor rezultate	- Chestionar scris - Răspuns oral - Caiet de laborator, lucrări experimentale, referate etc. - Demonstrație practică	$P_3=10\%$ $N_3 \geq 5$	
11.4d Proiect	Calitatea proiectului realizat, corectitudinea documentației proiectului, justificarea soluțiilor alese	- Autoevaluarea, prezentarea și/sau susținerea proiectului - Evaluarea critică a unui proiect	$P_4=30\%$ $N_4 \geq 5$	
11.5 Standard minim de performanță ²⁵			$N_T=5$	$P_T=100\%$
$N_T = 1 + 0,9 \times \sum_{n=1}^4 (P_n \times N_n) \geq 5$ $P_T = P_1 + P_2 + P_3 + P_4 = 100\%$ $N_T = 1 + 0,9 \times [(P_{1.1} \times N_{1.1} + P_{1.2} \times N_{1.2} + P_{1.3} \times N_{1.3} + P_{1.4} \times N_{1.4}) + P_2 \times N_2 + P_3 \times N_3 + P_4 \times N_4]$ Unde: 1 = punctul din oficiu (adăugat la calculul notei finale) P = Pondere (P_T = Pondera totală); N = Nota (N_T = Nota finală);				

Fișa disciplinei cuprinde componente adaptate persoanelor cu CES (persoane cu dizabilități și persoane cu potențial înalt), în funcție de tipul și gradul acestora, la nivelul tuturor elementelor curriculare (competențe, obiective, conținuturi, metode de predare, evaluare alternativă), pentru a asigura șanse echitabile în pregătirea academică a tuturor studenților, acordând atenție sporită nevoilor individuale de învățare.

Data completării: | 1 | 5 | / | 0 | 9 | / | 2 | 0 | 2 | 5 |

Data avizării în Departament:

0	1	/	1	0	/	2	0	2	5
---	---	---	---	---	---	---	---	---	---

	Grad didactic, titlul, prenume, numele	Semnătura
Titular disciplină	Lector univ. dr. Mircea-Adrian MUȘAN	
Responsabil program de studii	Lector univ. dr. Ioan-Daniel HUNYADI	
Director Departament	Profesor univ. dr. Mugur Alexandru ACU	

¹ Licență / Master

² 1-4 pentru licență, 1-2 pentru master

³ 1-8 pentru licență, 1-3 pentru master

⁴ Examen, colocviu sau VP A/R – din planul de învățământ

⁵ Regim disciplină: O=Disciplină obligatorie; A=Disciplină opțională; U=Facultativă

⁶ Categoria formativă: S=Specialitate; F=Fundamentală; C=Complementară; I=Asistată integral; P=Asistată parțial; N=Neasistată

⁷ Este egal cu 14 săptămâni x numărul de ore de la punctul 3.1 (similar pentru 3.2.a.b.c.d.e.)

⁸ Liniile de mai jos se referă la studiul individual; totalul se completează la punctul 3.37.

⁹ Între 7 și 14 ore

¹⁰ Între 2 și 6 ore

¹¹ Suma valorilor de pe liniile anterioare, care se referă la studiul individual.

¹² Suma (3.5.) dintre numărul de ore de activitate didactică directă (NOAD) și numărul de ore de studiu individual (NOSI) trebuie să fie egală cu numărul de credite alocate disciplinei (punctul 3.7) x nr. ore pe credit (3.6.)

¹³ Numărul de credit se calculează după formula următoare și se rotunjește la valori vecine întregi (fie prin micșorare fie prin majorare)

$$\text{Nr. credite} = \frac{\text{NOCpSpD} \times C_C + \text{NOApSpD} \times C_A}{\text{TOCpSdP} \times C_C + \text{TOApSdP} \times C_A} \times 30 \text{ credite}$$

Unde:

- NOCpSpD = Număr ore curs/săptămână/disciplina pentru care se calculează creditele
- NOApSpD = Număr ore aplicații (sem./lab./pro.)/săptămână/disciplina pentru care se calculează creditele
- TOCpSdP = Număr total ore curs/săptămână din plan
- TOApSdP = Număr total ore aplicații (sem./lab./pro.)/săptămână din plan
- C_C/C_A = Coeficienți curs/aplicații calculate conform tabelului

Coeficienți	Curs	Aplicații (S/L/P)
Licență	2	1
Master	2,5	1,5
Licență lb. străină	2,5	1,25

¹⁴ Se menționează disciplinele obligatoriu a fi promovate anterior sau echivalente

¹⁵ Tablă, videoproiector, flipchart, materiale didactice specifice, platforme on-line etc.

¹⁶ Tehnică de calcul, pachete software, standuri experimentale, platforme on-line etc.

¹⁷ Rezultatele învățării se vor menționa în funcție de standardele specifice ale Comisiilor de specialitate ARACIS (<https://www.aracis.ro/ghiduri/>)

¹⁸ Titluri de capitole și paragrafe

¹⁹ Expunere, prelegere, prezentare la tablă a problematicei studiate, utilizare videoproiector, discuții cu studenții (pentru fiecare capitol, dacă este cazul)

²⁰ Discuții, dezbateri, prezentare și/sau analiză de lucrări, rezolvare de exerciții și probleme

²¹ Legătura cu alte discipline, utilitatea disciplinei pe piața muncii

²² CPE – condiționează participarea la examen; nCPE – nu condiționează participarea la examen; CEF - condiționează evaluarea finală; N/A – nu se aplică

²³ Se va preciza numărul de teste și săptămânile în care vor fi susținute.

²⁴ Cercuri științifice, concursuri profesionale etc.

²⁵ Se particularizează la specificul disciplinei standardul minim de performanță din grila de competențe a programului de studii, dacă este cazul.