



FIȘA DISCIPLINEI ELEMENTE DE PROGRAMARE AVANSATE

1.1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA „OVIDIUS” DIN CONSTANȚA
1.2 Facultatea	Matematică și Informatică
1.3 Departamentul	Matematică și Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Master
1.6 Programul de studiu/Calificarea	Securitate cibernetică și învățare automată
1.7 Anul universitar	2025

2. Date despre disciplină

2.1 Denumirea disciplinei	Elemente avansate de programare					
2.2 Cod disciplina	CSML.1.1.03					
2.3 Titularul activitatilor de curs	Profesor Dorin-Mircea POPOVICI, PhD.					
2.4 Titularul activitatilor aplicative	Profesor Dorin-Mircea POPOVICI, PhD.					
2.5 Anul de studiu	1	2.6 Semestrul	1	2.7 Tipul de evaluare	E	2.8 Regimul disciplinei */** DF/DOB

* DF – disciplină fundamentală, DS – disciplină de specializare, DC – disciplină complementară

** DOB – disciplină obligatorie; DOP – disciplină opțională; DFA – Disciplină facultativă

3. Timpul total (ore pe semestru)

3.1 Număr de ore activități directe pe săptămână	3	din care: 3.2 curs	1	3.3 aplicații***	2
3.4 Total ore activități directe pe semestru	42	din care: 3.5 curs	14	3.6 aplicații	28
3.7 Total ore de studiu individual					83
Distribuția fondului de timp					[ore]
Studiul cărților, manualelor, suportului de curs, notițelor, bibliografie minimală recomandată					39
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					16
Pregătire seminar / laborator / proiect, teme, referate, portofolii și eseuri					20
Pregătire pentru prezentări sau verificări					0
Pregătire pentru examinarea finală					8
Alte activități: consultații					0
3.8 Total ore pe semestru	125				
3.9 Număr de credite	5				

*** S - seminar; L - laborator; P - proiect

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Licență - Programare orientată pe obiecte (opțional)
4.2 de rezultate ale învățării	Competențe de programare (obligatorii), programare orientată pe obiecte (opțional)



5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

5.1. de desfășurare a cursului	O sală de curs dotată cu proiector
5.2. de desfășurare a laboratorului*	Laborator de calculatoare

*Se alege tipul de aplicație aferent disciplinei

6. Obiectivele disciplinei

6.1 Obiectivul general al disciplinei	Studentul va fi familiarizat cu paradigma programării orientate pe obiecte.
6.2 Obiective specifice	Crearea și dezvoltarea competențelor necesare pentru a dezvolta aplicații complexe, scalabile și robuste, bazate pe paradigma programării orientate pe obiecte, utilizând platforma de dezvoltare Microsoft Visual Studio și unul dintre limbajele C++ / C# / Java / J#.

7. Rezultatele învățării

Cunoștințe	Studentul / Absolventul - înțelege principiile de codare sigură, vulnerabilitățile comune și cele mai bune practici în domeniul securității software.
Aptitudini	Studentul/Absolventul - dezvoltă aplicații software cu mecanisme de securitate integrate și verifică codul existent pentru a identifica vulnerabilitățile
Responsabilitate și autonomie	Studentul/absolventul: - asigură fiabilitatea și securitatea software-ului pe tot parcursul ciclului de dezvoltare și demonstrează responsabilitate pentru practicile de codare securizate.

8. Conținut

8.1 Curs	Metode de predare	Număr de ore
1. Introducere în programarea orientată pe obiecte Programarea orientată pe obiecte: definiții, clasificări, particularități.	Metode de predare interactive	2 ore
2. Paradigma orientată spre obiect Obiecte, clase (constructori, modificatori de acces, metode), ierarhii de clase, clase abstracte, moștenire, polimorfism, interfețe.	Dialog Problematizare Conversație	8 ore
3. Gestionarea erorilor prin excepții Ce este o excepție?, Detectarea și „captarea” unei excepții, blocul „Try”, excepții standard, crearea propriilor excepții.	Metode care contribuie la	2 ore



4. Sistemul I/O Intrări și ieșiri la nivel de octet, lucrul cu fișiere, fluxuri de date, analiză lexicală, serializarea obiectelor.	dezvoltarea gândirii critice Programe Învățare independentă și prin cooperare	2 ore
Bibliografie [1]. Bruce Eckel, <i>Thinking in Java</i> , ediția ^a 3-a, revizuirea 4.0. [2]. Jawahar Puvvala, Alok Pota, <i>.NET for Java Developers: Migrating to C#, Addison Wesley</i> , 2003, ISBN 0-672-32402-4. [3]. Irina Athanasu, et. al., <i>Limbajul Java – O perspectivă pragmatică</i> , Ed. a II-a, Ed. Agora, 2000. [4]. Galan C., <i>Curs de C#, Ed. L&S Info-mat</i> , 2008. [5]. Mayo J., <i>Microsoft Visual Studio 2010, A Beginner's Guide</i> , Ed. McGraw Hill, 2010. [6]. Dan C., <i>C# Object-oriented programmings</i> , Ed. Apress, 2011. [7]. Popovici D.M., Popovici I.M., <i>C++. Tehnologia orientate spre obiecte. Aplicatii</i> , Ed. Teora, Bucuresti, 2000		
8.2 Aplicații (laborator)* <i>* Tipul aplicației se alege în funcție de natura cursului</i>	Metode de predare	Număr de ore
1. Introducere în programarea orientată spre obiecte Programarea orientată pe obiecte: definiții, clasificări, particularități.	Metode de predare interactive	4 ore
2. Paradigma orientată spre obiect Obiecte, clase (constructori, modificatori de acces, metode), ierarhii de clase, clase abstracte, moștenire, polimorfism, interfețe.	Dialog Problematizare Conversație	16 ore
3. Gestionarea erorilor prin excepții Ce este o excepție?, Detectarea și „captarea” unei excepții, Blocul „Try”, Excepții standard, Crearea propriilor excepții.	Metode care contribuie la dezvoltarea gândirii critice	4 ore
4. Sistemul I/O Intrări și ieșiri la nivel de octet, lucrul cu fișiere, fluxuri de date, analiză lexicală, serializarea obiectelor.	Programe Învățare independentă și prin cooperare	4 ore
Bibliografie [1]. Bruce Eckel, <i>Thinking in Java</i> , ediția ^a 3-a, revizuirea 4.0. [2]. Jawahar Puvvala, Alok Pota, <i>.NET for Java Developers: Migrating to C#, Addison Wesley</i> , 2003, ISBN 0-672-32402-4. [3]. Irina Athanasu, et. al., <i>Limbajul Java – O perspectivă pragmatică</i> , Ed. a II-a, Ed. Agora, 2000. [4]. Galan C., <i>Curs de C#, Ed. L&S Info-mat</i> , 2008. [5]. Mayo J., <i>Microsoft Visual Studio 2010, A Beginner's Guide</i> , Ed. McGraw Hill, 2010. [6]. Dan C., <i>C# Object-oriented programmings</i> , Ed. Apress, 2011. [7]. Popovici D.M., Popovici I.M., <i>C++. Tehnologia orientate spre obiecte. Aplicatii</i> , Ed. Teora, Bucuresti, 2000		



9. Evaluare

Tip de activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Participarea activă la activitățile cursului	Discuții	10
9.5 Aplicații* <i>*Tipul de aplicație va fi ales în funcție de natura cursului</i>	Crearea unui portofoliu de probleme de laborator	Evaluare orală	10
	Realizarea sarcinilor de laborator	Discuție	20
	Proiect semestrial	Evaluare orală	20
	Examen	Nota la examen	30
	Implicit		10
9.6 Standard minim de performanță / Condiții de promovare - Nota 5 din 10.			
Implementarea unei aplicații funcționale și corecte într-un limbaj orientat către un obiect ales liber, în conformitate cu cerințele exprimate de profesor în biletul de examen și citite, înțelese și acceptate de student prin participarea la examen.			

Data finalizării,
12.09.2025

Lector,
Nume/Prenume /Semnătură
Prof. Dorin-Mircea POPOVICI, PhD

Profesor îndrumător,
Nume/Prenume/Semnătură
Prof. Dorin-Mircea POPOVICI, PhD

Data aprobării la nivel de departament,
15.09.2025

Șef de departament,
Nume/Prenume/Semnătură
Prof. univ. dr. Elena PELICAN

Decan,
Nume/Prenume/Semnătură
Prof. univ. dr. Aurelian NICOLA