

**D.O.004 PROGRAMAREA ORIENTATĂ PE OBIECTE**
**1. Date despre disciplină**

<b>Facultatea</b>	Calculatoare, Informatică și Microelectronică				
<b>Departamentul</b>	Ingineria Software și Automatică				
<b>Ciclul de studii</b>	Studii superioare de licență, ciclul I				
<b>Programul de studii</b>	0613.3 Ingineria Software				
<b>Anul de studii</b>	<b>Semestrul</b>	<b>Tip de evaluare</b>	<b>Categoria formativă</b>	<b>Categoria de opționalitate</b>	<b>Credite ECTS</b>
Anul II ( <i>învățământ cu frecvență</i> )	III	E	D - Disciplina de domeniu profesional.	O - unitate de curs obligatorie	6

**2. Timpul total estimat**

Total ore în planul de învățământ	Din care				
	Ore auditoriale			Lucrul individual	
	Curs	Lucrări practice	Seminar	Studiul materialului teoretic	Proiectare
Învățământ cu frecvență	30	30/30	30/30	90	

**3. Precondiții de acces la disciplină**

<b>Conform planului de învățământ</b>	Programarea Calculatoarelor Structuri de date și algoritmi
<b>Conform competențelor</b>	Cunoștințe și abilități de concepere și elaborare a algoritmilor și programelor în limbajul C pentru rezolvarea problemelor la calculator

**4. Condiții de desfășurare a procesului educațional pentru**

<b>Curs</b>	Pentru prezentarea materialului teoretic în sala de curs este nevoie de proiector, PC/laptop și acces la internet. Nu vor fi tolerate întârzierile studenților, precum și convorbirile telefonice în timpul cursului
<b>Lucrări practice/seminare</b>	Studenții vor perfecta rapoarte conform condițiilor formulate în indicațiile metodice. Termenul de susținere a lucrării de laborator – o săptămână după finalizarea acesteia.

**5. Competențe specifice acumulate**

<b>Competențe profesionale</b>	CP2. Proiectarea și dezvoltarea aplicațiilor CP7. Ingineria sistemelor
<b>Competențe transversale</b>	CT1. Aplicarea principiilor, normelor și valorilor eticii profesionale. CT2. Identificarea, descrierea și derularea activităților organizate într-o echipă cu dezvoltarea capacităților de comunicare și colaborare, dar și cu asumarea diferitelor roluri (de execuție și conducere). CT3. Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea propriilor cunoștințe profesionale, economice și de cultura organizațională.

**6. Obiectivele disciplinei**

<b>Obiectivul general</b>	Insușirea de către studenți a noțiunilor, conceptelor și exemplelor din programarea orientată obiect, Familiarizarea studenților cu tehnici de bază specifice programării orientate obiect, Construcția și analiza unor algoritmi specifici programării orientate obiect.
<b>Obiectivele specifice</b>	La finalizarea cu succes a acestei discipline, studenții vor fi capabili să: utilizeze tehnicile de programare orientată pe obiecte (POO); implementeze în Java/C++ tehnici ale POO; realizeze aplicații bazate pe POO.

**7. Conținutul disciplinei**

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
<b>Tematica cursurilor</b>	
T 1. Paradigme de programare. Programarea procedurală, funcțională și Concepte de bază ale POO.	2
T 2. Conceptele la baza POO. POO ca metodă de Structurare, impunerea Fiabilității, tehnici de Clasificare, și de Epistemologie.	2
T 3. Clasa: definirea și componentele claselor. Funcții speciale constructor și destructor.	2
T 4. Încapsularea și modificatori de acces: implicit, privat, public.	2
T 5. Moștenirea. derivarea simplă și multiplă a claselor. Modificator de acces: protejat.	2
T 6. Relații între clase: asociere, agregare, compoziție. Componente cuplate și decuplate în organizarea aplicațiilor POO.	2
T 7. Interfețe și tipuri de date abstracte. Inversarea controlului și inversarea dependențelor.	2
T 8. Polimorfismul static și dinamic.	2
T 9. Polimorfismul static: supraîncărcarea metodelor/funcțiilor și operatorilor.	2
T 10. Polimorfismul dinamic: subtip, șabloane, uniuni și tipuri de date algebrice.	2
T 11. Programarea orientată pe date în POO. Clase de date/record, imutabilitate datelor, interfețe sigilate.	2
T 12. Criterii de calitate a aplicațiilor POO interni: citibil, modular, structurare.	2
T 13. Criterii de calitate a aplicațiilor POO externi: corectitudine, robustețe, extensibilitate, reutilizabilitate, timp de răspuns, compatibilitate, eficiență, funcționalitate și ușurința utilizării.	2
T 14. Indici de probleme în aplicațiile POO și Datoria Tehnică. Tehnici comune de refactorizare a aplicațiilor POO.	2
T 15. Principii și aplicarea practică a principiilor S.O.L.I.D în aplicațiile POO.	2
<b>Total curs:</b>	<b>30</b>
<b>Tematica seminarelor</b>	
S 1. Facilitățile limbajului de programare C++ și Java. Primitive și referințe.	2
S 2. Controlul Fluxului și modularitatea claselor/metodelor.	2
S 3. Utilizarea *this *super. Metode și variabile statice.	2
S 4. Programarea orientată pe date. Interfețe sigilate, date *record/*data class. Date imutabile.	2
S 5. Tehnici eficiente de depanare pentru identificarea problemelor.	2
S 6. Utilizarea instrumentelor de debugging.	2
S 7. Clase container. Iteratorii. Algoritmi.	4
S 8. Testarea aplicațiilor POO.	2
S 9. Programarea declarativă și imperativă.	2
S 10. Integrarea tehnicilor din programare funcțională în POO. Funcții anonime.	4
S 11. Tratarea excepțiilor predefinite. Excepții personalizate.	2
S 12. Șabloane funcții și șabloane clase.	2
S 13. Aplicarea conceptelor OOP	2
<b>Total seminare:</b>	<b>30</b>
<b>Tematica practicilor</b>	
P1. Implementarea conceptelor de bază OOP. Modificatori de acces, deserializarea, prelucrearea și serializarea datelor din fișiere.	7
P2. Descrierea claselor. Relații de clase și interacțiunea obiectelor. Polimorfism static. Relații de compoziție.	7

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
P3. Moștenire și ierarhii. Aplicarea polimorfismului dinamic. Reutilizarea prin moștenire. Conversii *upcasting și *downcasting ale obiectelor. Structura aplicațiilor stratificate.	8
P4. Interfețe și polimorfism. Agregare și inversarea dependențelor. Dezvoltarea testelor pentru testarea aplicației. Tehnici de programarea sarcinilor concomitente în aplicațiile POO.	8
<b>Total practice:</b>	<b>30</b>

## 8. Referințe bibliografice

<b>Principale</b>	<ul style="list-style-type: none"> <li>• Bruce Eckel “Thinking in C++”, 2000</li> <li>• Kathy Sierra, Bert Bates, Trisha Gee “Head First Java” 3<sup>rd</sup> edition, 2022</li> <li>• Bertrand Meyer “Object Oriented Software Construction”, 1977</li> <li>• Heileman Gregory L. Data Structures, Algorithms and Object Oriented Programming. – McGraw – Hill, 1996.</li> <li>• Joshua Bloch, “Effective Java” 3<sup>rd</sup> edition, 2001</li> </ul>
<b>Suplimentare</b>	<ul style="list-style-type: none"> <li>• Bartosz Milewski “C++ In Action. Industrial-Strength Programming Techniques”</li> <li>• Kris Jamsa si Lars Klander, Totul despre C si C++ Manualul fundamental de programare in C si C++, Ed. Teora, (traducere 2007);</li> <li>• Gîncu S. Metodologia rezolvării problemelor de informatică în stilul orientat pe obiecte, Chișinău, 2012, 112 p.</li> </ul>

## 9. Utilizarea IA generativă

<b>Permișiunea de utilizare</b>	<p>Utilizarea IA generative în cadrul temelor și proiectelor este permisă, cu condiția ca studenții să respecte următoarele reguli:</p> <ul style="list-style-type: none"> <li>• IA generativă poate fi utilizată pentru generarea de idei, structuri de text sau cod, dar toate materialele generate trebuie să fie revizuite și ajustate de către student pentru a se asigura că acestea corespund cerințelor academice.</li> <li>• Orice utilizare a IA generative trebuie să fie declarată în secțiunea de apendice a fiecărei lucrări, folosind fraza: "În timpul pregătirii acestei lucrări, autorul a utilizat [NUME INSTRUMENT / SERVICIU] în scopul [MOTIV]. După utilizarea acestui instrument/serviciu, autorul a revizuit și editat conținutul după cum a fost necesar și își asumă întreaga responsabilitate pentru conținutul lucrării."</li> </ul>
<b>Restricții de utilizare</b>	<p>Studenții nu trebuie să considere IA generativă ca o sursă de încredere pentru informații, deoarece nu oferă referințe clare sau surse documentate.</p> <ul style="list-style-type: none"> <li>• Nu este permisă citarea directă a conținutului generat de IA în lucrările academice ca și cum ar fi sursă primară.</li> <li>• Activitățile în care este interzis utilizarea IA generativă sunt specificare de profesor și sunt de regulă evaluări intermediare și finale sau care nu presupun activități de dezvoltare a competențelor profesionale.</li> </ul>

## 10. Evaluare

Periodică		Curentă	Proiect	Examen
EP 1	EP 2			
<b>Învățământ cu frecvență</b>				
10%	10%	10%	30%	40%

Standard minim de performanță. Prezența și activitatea la prelegeri și lucrări practice. Prezentarea proiectului de an. Obținerea notei minime de „5” la fiecare lucrări practice și proiectul de an.

### 11. Criterii de evaluare

Activitate	Componente evaluare	Metodă de evaluare, Criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
<b>Învățământ cu frecvență</b>				
<b>Evaluare periodică I</b>	Conținut teoretic, teme 1-5	Test pe MOODLE	100%	<b>10%</b>
<b>Evaluare periodică II</b>	Conținut teoretic, teme 6-10	Test pe MOODLE	100%	<b>10%</b>
<b>Evaluare curentă</b>	Activitatea practică	Discuții în cadrul orelor de practică	50%	<b>10%</b>
		Raport pentru fiecare lucrare de practică încărcat pe MOODLE	50%	
<b>Lucrul individual/Proiect de an</b>	Cercetare la temă	Referat/Prezentare/discurs public. Raportul încărcat pe MOODLE	100%	<b>30%</b>
<b>Evaluarea finală</b>	Conținut teoretic și practic	Test pe MOODLE	100%	<b>40%</b>
<b>Învățământ cu frecvență redusă</b>				
<b>Evaluare curentă și periodică</b>	Activitatea practică	Raport pentru fiecare lucrare practică încărcat pe MOODLE	100%	<b>20%</b>
<b>Lucrul individual/Proiect de an</b>	Cercetare la temă	Referat/Prezentare/discurs public. Raportul încărcat pe MOODLE	100%	<b>30%</b>
<b>Evaluarea finală</b>	Conținut teoretic și practic	Test pe MOODLE	100%	<b>50%</b>