

FIȘA DISCIPLINEI

PROGRAMARE ORIENTATA OBIECT

1. Date despre program

1.1. Instituția de învățământ	Universitatea „1 Decembrie 1918” din Alba Iulia
1.2. Facultatea	de Informatica si Inginerie
1.3. Departamentul	de Informatica, Matematica si Electronica
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Licență
1.6. Programul de studii	Informatică ESCO-08: 2512/ Software developers Cod COR: Analist/251201, Programator de sistem informatic/251204, Inginer de sistem în informatică/251203

2. Date despre disciplină

2.1. Denumirea disciplinei		Programare orientata obiect			2.2. Cod disciplină		INFO 204				
2.3. Titularul activității de curs				Rotar Corina							
2.4. Titularul activității de seminar				Căpîlnas Matei/ Munteanu Bardan Raul Octav							
2.5. Anul de studiu		II	2.6. Semestrul		I	2.7. Tipul de evaluare (E/C/VP)		E	2.8. Regimul disciplinei (O – obligatorie, Op – opțională, F – facultativă)		O

3. Timpul total estimat

3.1. Numar ore pe saptamana	5	din care: 3.2. curs	2	3.3. seminar/laborator	3
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6. seminar/laborator	42
Distribuția fondului de timp					Ore
Studiul după manual, suport de curs, bibliografie și notițe					28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					12
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					10
Tutoriat					-
Examinări					5
Alte activități					-

3.7 Total ore studiu individual	55
3.8 Total ore din planul de învățământ	70
3.9 Total ore pe semestru	125
3.10 Numărul de credite	5

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	1. Fundamentele programarii
4.2. de competențe	- C1.1 Descrierea adecvată a - paradigmelor de programare și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic

	- C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.
--	---

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	<i>Sala dotata cu Videoproiector/tabla</i>
5.2. de desfășurarea a seminarului/laboratorului	<i>Laborator specific/ calculatoare/ software</i>

6. Competențe specifice acumulate

Competențe profesionale	<i>CP1(1 ECTS) analizeaza procese de afaceri</i> <i>CP2(1 ECTS) elaboreaza documentatie în conformitate cu cerintele legale</i> <i>CP24(1 ECTS) utilizeaza metodologii de proiectare dirijata de utilizator</i> <i>CP29(1 ECTS) utilizeaza sabloane de proiectare de software</i> <i>CP32(1 ECTS) interpreteaza cerinte tehnice</i>
Competențe transversale	Nu e cazul

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

7.1 Obiectivul general al disciplinei	<i>Dezvoltarea capacității studentului de a dezvolta aplicații software dedicate rezolvării problemelor. Dezvoltarea abilităților de a concepe tipuri abstracte de date și bibliotecile aferente. Crearea unui stil de programare orientat obiect riguros și eficient</i>
7.2 Obiectivele specifice	<i>Dezvoltarea abilitatii studentului de a gestiona eficient informatiile prin tipuri abstracte de date/clase si de a concepe in mod riguros metode de exploatare a acestora. Intocmirea unei documentatii coerente pe marginea aplicațiilor de complexitate medie.</i>

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Paradigma programarii orientate obiect. Concepte de baza.	<i>Prelegere, discutii</i>	
2. Programarea prin abstractizarea datelor. Particularitati C++.	<i>Prelegere, discutii</i>	
3. Clase si obiecte. Date si functii membre. Specificatori de acces	<i>Prelegere, discutii</i>	
4. Constructori. Desctorori. Constructorul de copiere	<i>Prelegere, discutii</i>	
5. Elemente statice si const ale claselor.	<i>Prelegere, discutii</i>	
6. Elemente friend. Supraincercarea operatorilor binari.	<i>Prelegere, discutii</i>	
7. Supraincercarea operatorilor (II).	<i>Prelegere, discutii</i>	
8. Conversii.	<i>Prelegere, discutii</i>	
9. Clase derivate,Clase de baza.	<i>Prelegere, discutii</i>	
10. Mostenirea. Mostenirea multipla. Constructori, destructori si mostenire	<i>Prelegere, discutii</i>	
11. Funcții virtuale	<i>Prelegere, discutii</i>	
12. Polimorfismul.	<i>Prelegere, discutii</i>	
13. Clase generice.		

14. Exceptii. Intrari iesiri standard.		
8.2 Bibliografie 1. Bruce Eckel, Thinking in C++, manual online. 2. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997. 3. H. Schildt: C++ manual complet, Teora, 2000. 4. Peter Muller: Introduction to Object-Oriented Programming Using C++, resurse electronice.		
Seminar-laborator		
Instruire NTSM. Considerații de specificare și implementare a POO	Lucrare practica de laborator	
Clasa ca tip abstract de date. Implemenatarea clasei in C++.	Lucrare practica de laborator	
Clasa. Stuctură. Componente: attribute, metode. Exemple	Lucrare practica de laborator	
Domeniul public, private, protected. Exemple de aplicații.	Lucrare practica de laborator	
Constructorii și destructorii. Aplicații.	Lucrare practica de laborator	
Operatori. Supraîncărcarea operatorilor.	Lucrare practica de laborator	
Utilizarea mediilor vizuale în POO. Visual Studio .NET, C#. Aplicații consolă.	Lucrare practica de laborator	
Clase standard și clase utilizator. Definirea claselor în C#.	Lucrare practica de laborator	
Mostenirea. Clase friend. Exemple în C++ comparativ cu C#.	Lucrare practica de laborator	
Metode statice și virtuale. Legare statică și dinamică. Proiectarea și implementarea metodelor virtuale.	Lucrare practica de laborator	
Programarea aplicațiilor Windows folosind clase predefinite în C#.	Lucrare practica de laborator	
Polimorfism. Exemple.	Lucrare practica de laborator	
Bibliografie 1. Bruce Eckel, Thinking in C++, manual online. 2. Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997. 3. H. Schildt: C++ manual complet, Teora, 2000. 4. Peter Muller: Introduction to Object-Oriented Programming Using C++, resurse electronice.		

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

- *Nu e cazul. Disciplina Programare orientata obiect este o disciplina fundamentala obligatorie in planul de invatamant al specializarii Informatica. Conținutul disciplinei este conceput in scopul formarii si dezvoltării gândirii algoritmice a studentului din anul II.*

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<i>Evaluare finala</i>	<i>Examen scris</i>	60%

	-	-	-
10.5 Seminar/laborator	<i>Verificare pe parcurs</i>	<i>Portofoliu de lucrări practice de laborator</i>	40%
	-		-
10.6 Standard minim de performanță: Obținerea notei minime 5.			
Implementarea și documentarea de unități de program în limbaje de programare orientate obiect și folosirea eficientă a mediilor de programare. Definirea claselor, identificarea și implementarea relațiilor dintre acestea.			

Observatii: Recuperarea laboratoarelor se poate face in regim de consultații in timpul semestrului. De asemenea, in cazuri bine motivate, recuperarea orelor de laborator se mai poate face prin prezentarea de către student a portofoliului complet de lucrari practice - in ultima saptamana din semestrul II, in orele de consultații ale cadrului didactic titular.

Data completării Semnătura titularului de curs Semnătura titularului de seminar

17.09.2025

Data avizării în departament Semnătura director departament

23.09.2025

Data avizării în Consiliul Facultatii Semnătura decan

25.09.2025