

PROGRAMAREA ORIENTATĂ PE OBIECTE
1. Date despre disciplină

Facultatea	Calculatoare, Informatică și Microelectronică				
Departamentul	Ingineria Software și Automatică				
Ciclul de studii	Studii superioare de licență, ciclul I				
Programul de studii	0613.1 Tehnologia Informației				
Anul de studii	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
Anul II (<i>învățământ cu frecvență</i>)	III	E	D-Disciplina de domeniu profesional.	O - unitate de curs obligatorie	6
Anul II (<i>învățământ cu frecvență redusă</i>)	IV	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	Laborator/s eminar	Proiect de an	Studiul materialului teoretic	Pregătire aplicații
Învățământ cu frecvență	30	30/30		90	
Învățământ cu frecvență redusă	12	12/12		144	

3. Precondiții de acces la disciplină

Conform planului de învățământ	G.O.001 Programarea calculatoarelor F.O Structuri de date și algoritmi
Conform competențelor	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

Curs	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
Laborator/seminar	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

Competențe profesionale	CP2. Proiectarea și dezvoltarea aplicațiilor CP7. Ingineria sistemelor
Competențe transversale	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

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

7. Conținutul disciplinei

Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
Tematica cursurilor		
T1. Paradigme de programare. Concepte de bază ale POO	2	1
T 2. Facilități C++	2	
T 3. Definirea claselor. Funcții speciale Constructor și destructor	2	1
T 4. Particularități: clase și obiecte. Funcții și clase friend	2	1
T5-6. Supraîncărcarea operatorilor	4	1
T 7. Moștenire. Derivarea simplă a claselor	2	1
T 8. Funcții virtuale și polimorfism.	2	1
T 9. Moștenire. Derivarea multiplă a claselor	2	1
T 10. Ierarhii de clase	2	1
T 11. Programarea generică	2	1
T 12-13. Standard Template Library (STL)	4	1
T 14. Tratarea excepțiilor	2	1
T 15. Exemple de implementare a conceptelor POO	2	1
Total curs:	30	12
Tematica lucrărilor practice/seminarelor		
LP1. Structura – mecanism de abstractizare	2	1
LP2. Facilitățile limbajului de programare C++ vs limbajul C	2	
LP3. Clase și obiecte. Constructori și destructor pentru clasa	2	1
LP4. Funcții și Clase friend. Utilizarea *this și static în elaborarea programelor în stilul orientat pe obiecte	2	1
LP5. Supraîncărcarea operatorilor unari și binari	4	1
LP6. Realizarea moștenirii simple între două și mai multe clase	2	1
LP7. Polimorfism. Funcții virtuale. Clase virtuale	2	1
LP8. Moștenire multiplă. Clase abstracte. Ambiguități în relația de moștenire multiplă	2	1
LP9. Relații între clase. Asocierea. Agregarea. Compoziția.	2	1
LP10. Șabloane funcții și șabloane clase.	2	1
LP11. Containeri. Iteratorii. Algoritmii	4	1
LP12. Tratarea excepțiilor predefinite	2	1
LP13. Implementarea conceptelor POO	2	1
Total lucrări practice/seminare:	30	12
Tematica lucrărilor de laborator		
LL1. Structura – mecanism de abstractizare	2	1
LL2. Facilitățile limbajului de programare C++ vs limbajul C	2	
LL3. Clase și obiecte. Constructori și destructor pentru clasa	2	1
LL4. Funcții și Clase friend. Utilizarea *this și static în elaborarea programelor în stilul orientat pe obiecte	2	1
LL5. Supraîncărcarea operatorilor unari și binari	4	1
LL6. Realizarea moștenirii simple între două și mai multe clase	2	1
LL7. Polimorfism. Funcții virtuale. Clase virtuale	2	1
LL8. Moștenire multiplă. Clase abstracte. Ambiguități în relația de moștenire multiplă	2	1
LL9. Relații între clase. Asocierea. Agregarea. Compoziția.	2	1
LL10. Șabloane funcții și șabloane clase.	2	1
LL11. Containeri. Iteratorii. Algoritmii	4	1
LL12. Tratarea excepțiilor predefinite	2	1
LL13. Implementarea conceptelor POO	2	1
Total lucrări de laborator:	30	12

8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> 1. Microsoft Developer Network (MSDN) 2. Herbert Schildt, C++, Ed. Teora (traducere, 2002); 3. Kris Jamsa și Lars Klander, Totul despre C și C++ Manualul fundamental de programare în C și C++, Ed. Teora, (traducere 2007); 4. David Vandevoorde, Nicolai M. Josuttis “C++ Templates: The Complete Guide”. Addison Wesley, 2002 5. Heileman Gregory L. Data Structures, Algorithms and Object Oriented Programming. – McGraw – Hill, 1996.
Suplimentare	<ol style="list-style-type: none"> 1. Bruce Eckel “Thinking in C++”, 2000 2. Bartosz Milewski “C++ In Action. Industrial-Strength Programming Techniques” 3. Jeffrey Richter, Applied Microsoft .NET Framework Programming, Microsoft Press (2002) 4. Gîncu S. Metodologia rezolvării problemelor de informatică în stilul orientat pe obiecte, Chișinău, 2012, 112 p.

9. Evaluare

Periodică		Curentă	Studiu individual	Proiect/teză	Examen
EP 1	EP 2				
Învățământ cu frecvență					
15%	15%	15%	15%		40%
Învățământ cu frecvență redusă					
25%			25%		50%
Standard minim de performanță					

10. Criterii de evaluare

Activitate	Componente evaluare	Metodă de evaluare, Criterii de evaluare	Pondere în nota finală a activității	Ponderea în evaluarea disciplinei
Învățământ cu frecvență				
Evaluare periodică I	Conținut teoretic, teme 1-6	Test pe MOODLE	100%	15%
Evaluare periodică II	Conținut teoretic, teme 7-12	Test pe MOODLE	100%	15%
Evaluare curentă	Activitatea practică	Discuții în cadrul seminarelor	50%	15%
		Dosar completat cu Rapoarte	50%	
Studiul individual	Implementarea conceptelor POO	Produse program elaborate și prezentate	100%	15%
Evaluarea finală	Conținut teoretic și practic	Examen în scris. Notare conform baremului	100%	40%
Învățământ cu frecvență redusă				
Evaluare periodică I	Conținut teoretic, teme 1-6	Test pe MOODLE	40%	25%
Evaluare periodică II	Conținut teoretic, teme 7-12	Test pe MOODLE	40%	
Evaluare curentă	Activitatea practică	Dosar completat cu Rapoarte	20%	
Studiul individual	Implementarea conceptelor POO	Produse program elaborate și prezentate	100%	25%
Evaluarea finală	Conținut teoretic și practic	Examen în scris. Notare conform baremului	100%	50%