

PROGRAMAREA ORIENTATĂ PE OBIECTE

1. Date despre disciplină/modul

Facultatea	Calculatoare, Informatică și Microelectronică				
Departamentul	Informatică și ingineria sistemelor				
Ciclul de studii	Studii superioare de licență, ciclul I				
Programul de studiu	0612.1 Calculatoare și rețele				
Anul de studiu	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
II (învățământ cu frecvență)	III	E	D - de domeniu profesional	O - obligatorie	5
II (învățământ cu frecvență redusă)	IV				

2. Timpul total estimat

Total ore în planul de învățământ	Din care				
	Ore auditoriale		Lucrul individual		
	Curs	Laborator/seminar	Proiect de an	Studiul materialului teoretic	Pregătire aplicații
150 zi	30	30/15	-	45	30
150 fr	12	10/8	-	60	60

3. Precondiții de acces la disciplină/modul

Conform planului de învățământ	Programarea calculatoarelor, Structuri de date și algoritmi.
Conform competențelor	Obținerea deprinderilor practice de programare orientată pe obiecte și folosirea diverselor procedee de utilizare a obiectelor.

4. Condiții de desfășurare a procesului educațional pentru

Curs	Pentru prezentarea materialului teoretic în sala de curs este nevoie de proiector și calculator. Nu vor fi tolerate întârzierile studenților, precum și convorbirile telefonice în timpul cursului.
Laborator/seminar	Studenții vor perfecta rapoarte conform condițiilor impuse de indicațiile metodice. Termenul de predare a lucrării de laborator – până la următoarea lucrare de laborator. Pentru predarea cu întârziere a lucrării aceasta se depunceaază cu 1pct./ciclu de întârziere.

5. Competențe specifice acumulate

Competențe profesionale	<p>CPL 1. Proiectarea aplicațiilor</p> <p>K1 Tehnici de modelare a cerințelor și tehnici de analiză a nevoilor.</p> <p>K2 Metodele de dezvoltare a software-ului și argumentarea acestora (de exemplu, prototipuri, metode agile, retroinginerie etc.).</p> <p>K3 Metricile care se referă la dezvoltarea aplicațiilor.</p> <p>K4 Principiile de proiectare a interfeței pentru utilizator.</p> <p>K5 Limbajele pentru formalizarea specificațiilor funcționale.</p> <p>K6 Aplicațiile existente și arhitectura lor aferentă.</p> <p>K7 Sisteme de gestionare a bazelor de date (DBMS), depozite de date, informații de business etc.</p> <p>K8 Tehnologiile mobile.</p>
-------------------------	---

CPL 2. Proiectarea și dezvoltarea aplicațiilor

- K1 Programe/module software adecvate.
- K2 Componente hardware, instrumente și arhitecturi hardware.
- K3 Proiectarea funcțională și tehnică.
- K4 Tehnologiile de ultimă oră.
- K5 Limbaje de programare.
- K6 Baze de date (DBMS).
- K7 Sisteme de operare și platforme software.
- K8 Mediul de dezvoltare integrat (IDE - integrated development environment).
- K9 Dezvoltarea rapidă a aplicațiilor.
- K10 Problemele legate de drepturile de proprietate intelectuală (IPR).
- K11 Tehnologia de modelare tehnică și limbaje.
- K12 Limbajele de definire a interfeței (IDL).
- K13 Probleme de securitate.

CPL 3. Integrarea componentelor

- K1 Componente/module hardware/software, indiferent dacă sunt vechi, existente sau noi.
- K2 Impactul integrării unui sistem asupra organizației sau a sistemului existent.
- K3 Tehnici de interfațare între module, sisteme și componente.
- K4 Tehnici de testare a integrării.
- K5 Instrumentele de dezvoltare (ex. mediul de dezvoltare, gestionare, control al modificărilor și accesul la codul sursă).
- K6 Bune practici de design.

CPL 4. Testarea aplicațiilor

- K1 Tehnicile, infrastructura și instrumentele necesare utilizate în procesul de testare.
- K2 Ciclul de viață al unui proces de testare.
- K3 Tipurile de teste (funcțional, de integrare, performanță, utilizabilitate, sarcină etc.).
- K4 Standardele naționale și internaționale care definesc criteriile de calitate pentru testare.
- K5 Specificul tehnologiilor legate de web, cloud, instrumente mobile și de probleme de mediu.

CPL 5. Implementarea soluțiilor

- K1 Tehnici de analiză a performanței.
- K2 Tehnicile legate de gestionarea problemelor (funcționare, performanță, compatibilitate).
- K3 Software-ul de ambalare/packaging și metode și tehnici de distribuție/desfășurare.
- K4 Impactul implementării/ desfășurării asupra arhitecturii existente.
- K5 Tehnologiile și standardele care se utilizează în timpul implementării/ /desfășurării.

Competențe
transversale

CTL 1. Autonomie și responsabilitate

Demonstrează executarea responsabilă a sarcinilor profesionale în condiții de autonomie.

CTL 2. Interacțiune socială

Execută rolurile și activitățile specifice muncii în echipă și distribuie sarcinile între membri pe niveluri subordonate.

CTL 3. Dezvoltare personală și profesională

Conștientizează nevoia de formare continuă cu utilizarea eficientă a resurselor și tehnicilor de învățare pentru dezvoltarea personală și profesională.

6. Obiectivele unității disciplinei/modulului

Obiectivul general	Însușirea tehnicilor de programare structurată și de dezvoltare a programării POO. Învățarea unor tehnici de baza pentru realizarea programelor de mare fiabilitate și siguranță funcțională.
Obiectivele specifice	Să înțeleagă metodologiile de dezvoltare a programării orientate pe obiecte. Să formeze capacitatea de extragere, identificare și specificare a cerințelor. Să formeze capacitatea de proiectare orientată obiect a programelor. Să capete cunoștințe referitoare la identificarea etapelor de programare. Să aplice corect procedeele și metodele de implementare a programelor. Să capete cunoștințe și abilități în utilizarea obiectelor pentru diferite domenii.

7. Conținutul disciplinei/modulului

Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
Tematica prelegerilor		
T1. Introducere în POO. Paradigme de programare. Principiile POO. Introducere în limbajul de programare C++.	2	1
T2. Clase și obiecte. Accesul la membrii clasei. Constructorii și destructorul clasei. Pointerul this . Funcții friend și clase friend. Membrii statici ai claselor.	4	1
T3. Supraîncărcarea funcțiilor și operatorilor. Supraîncărcarea operatorilor binari. Supraîncărcarea operatorilor unari.	4	2
T4. Moștenirea și compoziția. Derivarea simplă a claselor. Derivarea multiplă a claselor. Clase virtuale. Polimorfismul, funcții virtuale și clase abstracte.	6	2
T5. Fluxuri și operații de intrare-ieșire (I/O). Fluxuri de date I/O standard și fluxurile I/O de tip fișier cu acces secvențial și direct. Fluxurile I/O în memorie.	4	2
T6. Funcții și clase generice. Programarea generică. Biblioteca de șabloane STL.	4	2
T7. Tratarea și prelucrarea excepțiilor. Identificarea tipului în timpul rulării (RTTI).Alte aspecte ale limbajului C++.	6	2
Total prelegeri:	30	12
Tematica lucrărilor practice		
LP1. Structura – mecanism de abstractizare.	2	1
LP2. Clase și obiecte. Constructori și destructor pentru clasa.	2	1
LP3. Supraîncărcarea operatorilor.	2	1
LP4. Moștenirea și compoziția.	2	1
LP5. Moștenirea multiplă. Clase virtuale.	2	1
LP6. Polimorfism. Funcții virtuale. Clase abstracte.	2	1
LP7. Funcții și clase generice. Biblioteca standard STL.	3	2
Total lucrări practice:	15	8
Tematica lucrărilor de laborator		
LL1. Structura – mecanism de abstractizare.	4	1
LL2. Clase și obiecte. Constructori și destructor pentru clasa.	4	1
LL3. Supraîncărcarea operatorilor.	4	1
LL4. Moștenirea și compoziția.	4	1
LL5. Moștenirea multiplă. Clase virtuale.	4	1
LL6. Polimorfism. Funcții virtuale. Clase abstracte.	4	2
LL7. Funcții și clase generice. Prezentarea rapoartelor.	6	3
Total lucrări de laborator:	30	10

8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> 1. Mirel Coșulschi, Octavian Mustafa, Programare în C++. Concepte moderne și aplicații, Editura Universitaria, 2015. 2. Introducere în programarea orientată-obiect : Concepte fundamentale din perspectiva ingineriei software / Mircea Cezar Preda, Ana-Maria Mirea, Doina Lavinia Preda, Constantin Teodorescu-Mihai ; coord.: Mircea Cezar Preda. – Iași : Polirom, 2010. –280 p. 3. Liviu Negrescu, Limbajele C și C++ pentru începători, Vol. II, (editia XI),Editura Albastra, Cluj-Napoca, 2005. 4. Kris Jamsa si Lars Klander, Totul despre C si C++ Manualul fundamental de programare in C si C++, Ed. Teora, (traducere 2007). 5. David Vandevoorde, Nicolai M. Josuttis "C++ Templates: The Complete Guide". Addison Wesley, 2002.
Suplimentare	<ol style="list-style-type: none"> 1. Bruce Eckel "Thinking in C++", 2000. 2. Vasile Stoicu-Tivadar, „Programare Orientata pe Obiecte”, Editura Orizonturi Universitare, Timisoara 2000. 3. Erich Gamma, Richard Helm, R. Johnson, J. Vlissides, „Design Patterns - Sabloane de proiectare”, Editura Teora, București, 2002. 4. L.A. Maciaszek, B.L. Liong - "Practical Software Engineering", Addison-Wesley, Pearson Education, 2005, ISBN 0-321-20465-4. 5. I. Sommerville - "Software Engineering", Addison-Wesley, Pearson Education, 8th Edition, 2006, ISBN 0-321-21026-3.

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

Prezența și activitatea la prelegeri, lecții practice și lucrări de laborator. Obținerea notei minime de „5” la fiecare dintre evaluări și lucrări de laborator. Obținerea notei minime de „5” la examen.