

### PROGRAMAREA ORIENTATĂ PE OBIECTE

#### 1. Date despre unitatea de curs/modul

<b>Facultatea</b>	Calculatoare, Informatică și Microelectronică				
<b>Departamentul</b>	Microelectronică și Inginerie Biomedicală				
<b>Ciclul de studii</b>	Studii superioare de licență, ciclul I				
<b>Programul de studiu</b>	0714.9 Inginerie biomedicală				
<b>Anul de studiu</b>	<b>Semestrul</b>	<b>Tip de evaluare</b>	<b>Categoria formativă</b>	<b>Categoria de opționalitate</b>	<b>Credite ECTS</b>
II (învățământ cu frecvență)	3	E	F – unitate de curs fundamentală	O - unitate de curs obligatorie	5

#### 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
	învățământ cu frecvență				
150	30	30/15	-	45	30

#### 3. Precondiții de acces la unitatea de curs/modul

Conform planului de învățământ	Matematica superioară, 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. Aceste competențe sunt formate de următoarele unități de curs, prevăzute de planul de învățământ: matematica superioară, programarea calculatoarelor, structuri de date și algoritmi etc.

#### 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 depuncea cu 1pct./ciclu de întârziere.

#### 5. Competențe specifice acumulate

Competențe profesionale	<p><b>CP2. Proiectarea sistemelor hardware, software și de comunicații:</b></p> <ul style="list-style-type: none"> <li>• Descrierea structurală și funcțională a componentelor hardware, software și de comunicații.</li> <li>• Explicarea și înțelegerea destinației, interacțiunii și funcționării componentelor hardware, software și de comunicații.</li> <li>• Elaborarea unor componente hardware, software și de comunicații folosind metode de proiectare, limbaje de programare și descriere hardware, algoritmi, structuri de date, protocoale și tehnologii.</li> <li>• Evaluarea caracteristicilor comportamentale și structurale ale sistemelor de calcul,</li> </ul>
-------------------------	--

	<p>aplicațiilor software și rețelelor de calculatoare în baza unor metrici.</p> <ul style="list-style-type: none"> <li>Proiectarea și implementarea componentelor, sistemelor de calcul de uz general și dedicate, aplicațiilor software și rețelelor de calculatoare.</li> </ul> <p><b>CP3.</b> Identificarea, formularea și soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor:</p> <ul style="list-style-type: none"> <li>Recunoașterea și descrierea unor tehnici și metode de rezolvare a sarcinilor de sinteză, modelare, simulare, verificare și implementare a echipamentelor, sistemelor de calcul, aplicațiilor software și rețelelor de calculatoare.</li> <li>Utilizarea adecvata a cunoștințelor interdisciplinare, a metodelor de soluționare și a mediilor de dezvoltare, efectuarea experimentelor și interpretarea rezultatelor.</li> <li>Aplicarea metodelor și tehnicilor de soluționare a problemelor din domeniu, utilizând unelte moderne de proiectare asistată de calculator.</li> <li>Evaluarea comparativă a performanțelor sistemelor de calcul, aplicațiilor software și rețelelor de calculatoare, utilizând instrumente alternative de analiză, în scopul optimizării performanțelor.</li> <li>Dezvoltarea și implementarea soluțiilor informatice pentru problem concrete utilizând tehnici CAD și CAE.</li> <li>Descrierea procedeeelor, tehnicilor și metodelor de bază necesare pentru exploatarea și dezvoltarea software.</li> <li>Explicarea funcționării și a interacțiunii cu mediul a aplicațiilor software</li> <li>Utilizarea unor metode specializate pentru configurarea și dezvoltarea aplicațiilor software</li> <li>Evaluarea calitativă și cantitativă a aplicațiilor software.</li> <li>Elaborarea produselor program utilizând metode și instrumente de lucru pentru proiectarea, integrarea și testarea componentelor și aplicațiilor software.</li> </ul>
Competențe transversale	<ul style="list-style-type: none"> <li>Demonstrarea capacității de lucru în echipă, identificarea rolurilor și responsabilităților individuale și comune, luarea deciziilor și atribuirea de sarcini, cu aplicarea de tehnici de relaționare și muncă eficientă în cadrul echipei</li> <li>Demonstrarea spiritului de inițiativă și acțiune pentru dezvoltarea profesională și personală, prin formare continuă utilizând surse de documentare în limba română și în limbile de circulație internațională</li> </ul>

### 6. Obiectivele unității de curs/modulului

Obiectivul general	Înșușirea 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	<p>Să înțeleagă metodologiile de dezvoltare a programării orientate pe obiecte.</p> <p>Să formeze capacitatea de extragere, identificare și specificare a cerințelor.</p> <p>Să formeze capacitatea de proiectare orientată obiect a programelor.</p> <p>Să capete cunoștințe referitoare la identificarea etapelor de programare.</p> <p>Să aplice corect procedeele și metodele de implementare a programelor.</p> <p>Să capete cunoștințe și abilități în utilizarea obiectelor pentru diferite domenii.</p> <p>Să aplice corect procedeele de management al proiectului, îmbunătățire a procesului de dezvoltare.</p>

### 7. Conținutul unității de curs/modulului

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
Tematica prelegerilor	
T1. Introducere în POO. Paradigme de programare. Principiile POO. Introducere în	2

limbajul de programare C++.	
T2. <b>Clase și obiecte.</b> Accesul la membrii clasei. Constructorii și destructorul clasei. Pointerul <b>this</b> . Funcții friend și clase friend. Membrii statici ai claselor.	4
T3. <b>Supraîncărcarea funcțiilor și operatorilor.</b> Supraîncărcarea operatorilor binari. Supraîncărcarea operatorilor unari.	4
T4. <b>Moștenirea și compoziția.</b> Derivarea simplă a claselor. Derivarea multiplă a claselor. Clase virtuale. Polimorfismul, funcții virtuale și clase abstracte. Relații între clase: compoziție (agregare), asociere și dependență.	6
T5. <b>Fluxuri și operații de intrare-ieșire (I/O).</b> Fluxuri de date I/O standard și fluxurile I/O de tip fișier cu acces secvențial și direct. Fluxurile I/O în memorie.	4
T6. <b>Funcții și clase generice.</b> Programarea generică. Biblioteca de șabloane STL.	4
T7. <b>Tratarea și prelucrarea excepțiilor.</b> Identificarea tipului în timpul rulării (RTTI). Alte aspecte ale limbajului C++.	6
<b>Total prelegeri:</b>	<b>30</b>
Tematica lucrărilor practice (seminarelor)	
LP1. Structura – mecanism de abstractizare.	2
LP2. Clase și obiecte. Constructori și destructor pentru clasa.	2
LP3. Supraîncărcarea operatorilor.	2
LP4. Moștenirea și compoziția. Polimorfismul.	2
LP5. Fluxurile In/Out standard și definite de utilizatori.	2
LP6. Funcții și clase generice. Biblioteca standard STL.	3
LP7. Prelucrarea excepțiilor. Blocurile try, throw, catch.	2
<b>Total lucrări practice:</b>	<b>15</b>
Tematica lucrărilor de laborator	
LL1. Structura – mecanism de abstractizare.	4
LL2. Clase (constructori și destructori). Funcții și clase prietene.	4
LL3. Supraîncărcarea operatorilor.	4
LL4. Moștenirea și compoziția.	4
LL5. Fluxurile In/Out standard și definite de utilizatori.	4
LL6. Templates: Template-uri pentru clase și funcții.	4
LL7. Prelucrarea excepțiilor. Blocurile try, throw, catch. Prezentarea rapoartelor.	6
<b>Total lucrări de laborator:</b>	<b>30</b>

### 8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> <li>1. Vasile Stoicu-Tivadar, „Programare Orientata pe Obiecte”, Editura Orizonturi Universitare, Timisoara 2000.</li> <li>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.</li> <li>3. Herbert Schildt, C++, Ed.Teora (traducere, 2002);</li> <li>4. Kris Jamsa si Lars Klander, Totul despre C si C++ Manualul fundamental de programare in C si C++, Ed. Teora, (traducere 2007).</li> <li>5. David Vandevoorde, Nicolai M. Josuttis “C++ Templates: The Complete Guide”. Addison Wesley, 2002.</li> </ol>
Suplimentare	<ol style="list-style-type: none"> <li>1. Bruce Eckel “Thinking in C++”, 2000.</li> <li>2. Erich Gamma, Richard Helm, R. Johnson, J. Vlissides, „Design Patterns - Sabloane de proiectare”, Editura Teora, București, 2002.</li> <li>3. L.A. Maciaszek, B.L. Liang - "Practical Software Engineering", Addison-Wesley, Pearson Education, 2005, ISBN 0-321-20465-4.</li> <li>4. I. Sommerville - "Software Engineering", Addison-Wesley, Pearson Education, 8th Edition, 2006, ISBN 0-321-21026-3.</li> </ol>

### 9. Evaluare

Forma de învățământ	Periodică		Curentă	Lucrul individual	Examen final
	Atestarea 1	Atestarea 2			
Cu frecvență	15%	15%	15%	15%	40%
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					

**Titularul unității de curs: conf. univ., dr. Mihail Kulev**  
**26.08.2020**