

MD-2045, CHIȘINĂU, STR. STUDENȚILOR, 9/4, TEL: 022 50-99-15, www.calc.fcim.utm.md
F.03.O.010 PROGRAMAREA ORIENTATĂ PE OBIECTE
1. Date despre unitatea de curs/modul

Facultatea	Calculatoare, Informatică și Microelectronică				
Departamentul	Informatică și Ingineria Sistemelor				
Ciclul de studii	Studii superioare de licență, ciclul I				
Programul de studiu	0.714.9 Inginerie biomedicală				
Anul de studiu	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
II (învățământ cu frecvență);	3;	E	F – unitate de curs fundamental	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	45	30/15	-	45	30

3. Precondiții de acces la unitatea de curs/modul

Conform planului de învățământ	Studentii trebuie să posedă abilități în matematica superioară și fizică, să cunoască un limbaj de programare de nivel înalt, să poată utiliza sistemele de operare uzuale.
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 și aplicată, 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	Studentii vor perfecta rapoarte conform condițiilor impuse de indicațiile metodice. Termenul de predare a lucrării de laborator – pînă la următoarea lucrare de laborator. Pentru predarea cu întârziere a lucrării aceasta se depunează cu 1pct./ciclu de întârziere.

5. Competențe specifice acumulate

Competențe profesionale	CP2. Proiectarea sistemelor hardware, software și de comunicații: <ul style="list-style-type: none"> ✓ Descrierea structurală și funcțională a componentelor hardware, software și de comunicații. ✓ Explicarea și înțelegerea destinației, interacțiunii și funcționării componentelor hardware, software și de comunicații. ✓ 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. ✓ Evaluarea caracteristicilor comportamentale și structurale ale sistemelor de calcul, aplicațiilor software și rețelelor de calculatoare în baza unor metrici. Proiectarea și implementarea componentelor, sistemelor de calcul de uz general și dedicate, aplicațiilor software și rețelelor de calculatoare.
-------------------------	--

	<p>CP3. Identificarea, formularea și soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor:</p> <ul style="list-style-type: none"> ✓ 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. ✓ Utilizarea adecvata a cunoștințelor interdisciplinare, a metodelor de soluționare și a mediilor de dezvoltare, efectuarea experimentelor și interpretarea rezultatelor. ✓ Aplicarea metodelor și tehnicilor de soluționare a problemelor din domeniu, utilizând unelte moderne de proiectare asistată de calculator. ✓ Evaluarea comparativă a performanțelor sistemelor de calcul, aplicațiilor software și rețelelor de calculatoare, utilizând instrumente alternative de analiză, în scopul optimizării performanțelor. <p>Dezvoltarea și implementarea soluțiilor informatice pentru probleme concrete utilizând tehnici CAD și CAE</p> <p>CP6. Utilizarea, configurarea și dezvoltarea sistemelor și aplicațiilor software</p> <ul style="list-style-type: none"> ✓ Descrierea procedeeleor, tehnicilor și metodelor de bază necesare pentru exploatarea și dezvoltarea software ✓ Explicarea funcționării și a interacțiunii cu mediul a aplicațiilor software ✓ Utilizarea unor metode specializate pentru configurarea și dezvoltarea aplicațiilor software ✓ Evaluarea calitativă și cantitativă a aplicațiilor software ✓ Elaborarea produselor program utilizând metode și instrumente de lucru pentru proiectarea, integrarea și testarea componentelor și aplicațiilor software
--	--

Competențe transversale	<ul style="list-style-type: none"> - 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 - 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
-------------------------	--

6. Obiectivele unității de curs/modulului

Obiectivul general	Înșușirea tehnicilor de programare structurată și de dezvoltare a programării concurente, prin rafinare în pași succesivi. 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 Apariția programării orientate pe obiecte. Motivarea necesității studierii principiilor de POO. Gândirea în termenii obiectelor. Clasele și obiectele. Ierarhia claselor și moștenirea. Legarea și redefinirea metodelor. Lămurire intuitivă a mecanismelor. Creșterea neliniară a complexității. Mecanismele de abstractizare.	2

T2. Încapsulare. Definiția. Tipurile claselor. Interfața și implementarea. Ascunderea informațiilor. Clasele și metodele în diferite limbaje de programare orientate pe obiecte (C++, Java). Mesajele, obiecte-exemplare, inițializare. Sintaxa transmiterii mesajului în limbaje diferite. Metodele de creare și inițializare (alocare automată, stivă sau heap). Constructorii. Tipurile constructorilor. Realizarea constructorilor în diferite limbaje. (C++, Java)	4
T3. Descriere a moștenirii. Subclasă, subtip și principiul de substituție. Subtipuri și controlul strict a tipurilor de date. Compoziția. Compoziția în diferite limbaje de programare. Sintaxa și realizarea compoziției. Moștenire în diferite limbaje de programare. (C++, Java).	8
T4. Moștenirea multiplă în diferite limbaje de programare. Utilizare interfețelor la realizarea moștenirii multiple.	2
T5. Clase interne. Clase interne în metode. Legătura cu clasele externe. Clase incluse. Accesul la obiecte din clasele incluse.	2
T6. Polimorfism în limbajele de programare. Mecanismele. Tipurile Redefinirea. Polimorfism pur. Funcții și clasele generice.	2
T7. Excepțiile și prelucrarea error-ilor. Prelucrarea excepțiilor. Crearea excepțiilor în diferite limbaje de programare. Excepțiile standarte.	2
T8. Introducerea și extragerea datelor prin flex. Flexuri de introducere. Fluxuri de extragere. Canale. Clase speciale. Modalități de citire și scriere a fișierelor. Citire scriere standart. Citire scriere specială. Buferizarea. Bufere. Manipularea datelor.	8
T9. Arhivarea datelor. Arhivarea standartă. Arhivatori a limbajelor de programare.	2
T10. Crearea apletelor. Avantajele apletelor. Activarea apletelor. Testarea apletelor.	2
T11. Colecții de obiecte. Tablouri. Containere. Iteratori. Colecții și liste. Mulțimi. Cărți cu date.	5
T12. Analiza și planificarea. Metodologia planificării. Strategii de analiză. Localizarea problemelor. Testarea modulelor. Susținerea fiabilității codului.	4
T13. Localizarea problemelor. Testarea modulelor. Susținerea fiabilității codului.	2
Total prelegeri:	45

Tematica activităților didactice	Numărul de ore
	învățământ cu frecvență
Tematica lucrărilor de laborator/seminarelor	
LL1. Clasele, obiectele, constructorii.	4
LL2. Redefinirea funcțiilor.	4
LL3. Moștenirea.	4
LL4. Polimorfismul.	4
LL5. Crearea excepțiilor.	4
LL6. Programarea apletelor.	4
LL7. Programarea în mediu vizual a proiectelor.	6
Total lucrări de laborator/seminare:	30

8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> 1. Cristian Frasinescu Curs practic de Java. Curs electronic. 2011. 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 3. Logofătu, Doina. Algoritmi fundamentali în JAVA : aplicații / Doina Logofătu. – Iași : Polirom, 2007. – 371 p.
Suplimentare	<ol style="list-style-type: none"> 1. Booch Grady, “Object-Oriented Analysis and Design”, Redwood, 2004 2. Liberty Jess “Teach Yourself C++ in 21 Days”, SAMS Publishing, Indianapolis., 1994 3. Mustea Ioneț “Inițiere în C++”, Microinformatica, Cluj-Napoca, 1993

9. Evaluare

Curentă		Proiect de an	Examen final
Evaluarea 1	Evaluarea 2		
30%	30%		40%
Standard minim de performanță			
Prezența și activitatea la prelegeri și lucrări de laborator; Obținerea notei minime de „5” la fiecare dintre atestări și lucrări de laborator; Demonstrarea în lucrarea de examinare finală a cunoașterii condițiilor de aplicare a procedeeleor de modelare constructivă.			