

INGINERIA PROGRAMĂRII
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	0612.1 Calculatoare și rețele				
Anul de studiu	Semestrul	Tip de evaluare	Categoria formativă	Categoria de opționalitate	Credite ECTS
IV (învățământ cu frecvență); IV (învățământ cu frecvență redusă)	7; 7	E, PA	S – unitate de curs de specialitate	A - unitate de curs opțională	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
150	30	30/15	15	30	30
150	12	10/4	30	60	34

3. Precondiții de acces la unitatea de curs/modul

Conform planului de învățământ	Programarea calculatoarelor, Programarea în limbajul C++, Programarea orientată pe obiecte.
Conform competențelor	Obținerea deprinderilor practice de elaborare a diagramelor UML.

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 – o săptămână după finalizarea acesteia. Pentru predarea cu întârziere a lucrării aceasta se depunțează cu 1pct./săptămână de întârziere.

5. Competențe specifice acumulate

Competențe profesionale	CP2. Proiectarea sistemelor hardware, software și de comunicații ✓ Descrierea structurală și funcțională a componentelor software. ✓ Explicarea și înțelegerea destinației, interacțiunii și funcționării componentelor software. ✓ Elaborarea unor componente software și de comunicații folosind metode de proiectare, limbaje de programare, algoritmi, structuri de date, protocoale și tehnologii. ✓ Evaluarea caracteristicilor comportamentale și structurale ale aplicațiilor software în baza unor metrici. ✓ Proiectarea și implementarea componentelor aplicațiilor software
Competențe profesionale	CP4. Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații ✓ Identificarea și descrierea parametrilor de bază ale performanțelor și aplicațiilor software. ✓ Explicarea interacțiunii factorilor care determină parametrii de performanță a aplicațiilor software. ✓ Aplicarea de principii și metode de bază pentru determinarea, calcularea și optimizarea parametrilor de performanță a aplicațiilor software.

	<ul style="list-style-type: none"> ✓ Gestionarea ciclului de viață a aplicațiilor software în baza evaluării performanțelor. ✓ Dezvoltarea de aplicații pentru monitorizarea și controlul parametrilor de bază a sistemelor software pentru îmbunătățirea și optimizarea performanțelor.
Competențe profesionale	<p>CP6. Utilizarea, configurarea și dezvoltarea sistemelor și aplicațiilor software</p> <ul style="list-style-type: none"> ✓ Descrierea procedurilor, 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	<p>CT2. 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</p> <p>CT3. 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</p>

6. Obiectivele unității de curs/modulului

Obiectivul general	Înșușirea tehnicilor de programare structurată și de dezvoltare a programelor în etape, 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 programelor.</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 proiectare.</p> <p>Să aplice corect procedeele și metodele de implementare a programelor.</p> <p>Să capete cunoștințe și abilități în utilizarea limbajului UML pentru modelarea conceptuală a proiectului.</p> <p>Să capete cunoștințe și abilități în verificarea și validarea programelor, tipurile de testare.</p> <p>Să aplice corect procedeele de management al proiectului, estimare a costurilor, îmbunătățire a procesului de dezvoltare.</p>

7. Conținutul unității de curs/modulului

Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
Tematica prelegerilor		
T1. Introducere. Sisteme socio-tehnice. Sisteme critice. Caracteristica planului de studiu la Ingineria Programării, bibliografia. Obiectul de studiu al Ingineriei Programării. Modelele principale ale Ingineriei Programării. Importanța Ingineriei Programării în pregătirea inginerilor. Definierea sistemelor socio-tehnice și sistemelor critice. Diferența, rolul și importanța lor în societate.	1	1
T2. Procesele de dezvoltare. Managementul proiectului. Modelele de dezvoltare a programelor, iterațiile și activitățile. Noțiuni RUP (Rational Unified Process) și CASE (Computer-Aided Software Engineering). Dirijarea activităților. Planificarea și stabilirea orarului proiectului.	2	1
T3. Cerințele programului. Procesele de inginerie a cerințelor. Cerințe funcționale	2	1

și ne-funcționale. Cerințele utilizatorului. Cerințele sistemului. Specificații pentru Interfața utilizator. Documentul cerințelor programului. Extragerea și analiza cerințelor. Validarea cerințelor. Gestionarea cerințelor. Exemplu.		
T4. Modelele sistemului. Specificațiile sistemelor critice. Specificațiile formale. Modele de comportament. Modele de prezentare a datelor. Modele de prezentare a obiectelor. Modele structurale. Specificațiile de prezentare a riscurilor. Specificațiile de fiabilitate, siguranță și securitate. Specificațiile formale în procesele software. Specificațiile interfețelor sub-sistemelor. Exemplu.	2	1
T5. Conceptul arhitectural. Arhitectura sistemelor distribuite. Arhitectura aplicației. Deciziile conceptului arhitectural. Descompunerea modulară. Arhitecturi de referință. Arhitecturi de sisteme multi-procesor, client-server, distribuite, inter-organizaționale. Sisteme de procesare a datelor, a tranzacțiilor, a evenimentelor și a limbajului.	2	1
T.6. Conceptul orientat obiect. Conceptul real-time al sistemului. Obiecte și obiectele claselor. Procesul de concepere orientat obiect. Evoluția conceptului. Concepția sistemului. Sisteme real-time, de monitorizare și control, de colectare a datelor. Soluții de design.	2	1
T.7. Reutilizarea soft-ului. Ingineria programării bazată pe componente. Șabloane de concepție. Aplicații framework. Componente și modele de componente. Structura unui component.	2	1
T.8. Implementarea sistemelor critice. Evoluția programului. Procesele și programarea sigură. Toleranța față de erori. Dinamica evoluției programului. Mentenanța programului. Procesele de evoluție. Evoluția sistemelor legacy (de succesiune).	2	1
T.9. Verificarea și validarea. Testarea programelor. Validarea sistemelor critice. Planificarea verificării și validării. Inspecția programului. Verificarea automată. Testarea sistemului. Testarea componentelor. Conceperea Test-Case-urilor. Validarea fiabilității. Asigurarea siguranței și securității.	3	1
T.10. Îmbunătățirea procesului. Gestiunea configurării. Calitatea procesului și al produsului. Clasificarea, măsurarea, analiza, modelarea și schimbarea procesului. Planificarea gestionării configurării. Gestiunea modificărilor, versiunilor și a livrărilor. Construirea sistemului. Aplicații CASE pentru gestiunea configurației.	4	1
T.11. Ingineria securității. Conceptele de securitate. Gestiunea riscurilor de securitate. Conceperea securității. Vulnerabilitatea sistemului.	4	1
T.12. Ingineria produselor program orientată pe servicii și pe aspecte. Serviciile – componente reutilizabile. Ingineria serviciilor. Dezvoltarea programelor utilizând serviciile. Separarea rolurilor. Aspecte, punctele de intersecție și divergență. Ingineria produselor program orientat pe aspecte.	4	1
Total prelegeri:	30	12

Tematica activităților didactice	Numărul de ore	
	învățământ cu frecvență	învățământ cu frecvență redusă
Tematica lucrărilor de laborator/seminarelor		
LL1. Elemente ale programului Enterprise Architect. Diagrame de prezentare a concepției proiectului. Repartizarea temelor pentru lucrări de laborator.	8	2
LL2. Extragerea și analiza cerințelor conform temei personale. Crearea documentului de cerințe. Construirea diagramei Use-Case în baza cerințelor colectate.	8	4

LL3. Estimarea costurilor proiectului. Crearea planului de dezvoltare. Crearea modelului conceptual al proiectului. Crearea modelului de componente și deployment al programului.	8	4
LL4. Implementarea componentelor sistemului analizat.	6	0
S1. Cerințele programului. Procesele de inginerie a cerințelor. Cerințe funcționale și ne-funcționale. Cerințele utilizatorului. Cerințele sistemului. Specificații pentru Interfața utilizator. Documentul cerințelor programului. Extragerea și analiza cerințelor. Validarea cerințelor. Gestionarea cerințelor.	2	1
S2. Modelele sistemului. Specificațiile sistemelor critice. Specificațiile formale. Modele de comportament. Modele de prezentare a datelor. Modele de prezentare a obiectelor. Modele structurale.	2	1
S3. Conceptul orientat obiect. Obiecte și obiectele claselor. Procesul de concepere orientat obiect.	2	1
S4. Reutilizarea soft-ului. Ingineria programării bazată pe componente. Șabloane de concepție. Aplicații framework. Componente și modele de componente. Structura unui component.	2	0
S5. Implementarea sistemelor critice. Evoluția programului. Procesele și programarea sigură. Toleranța față de erori. Dinamica evoluției programului. Mentenanța programului. Procesele de evoluție. Evoluția sistemelor legacy (de succesiune).	2	0
S6. Testarea programelor. Planificarea verificării și validării. Inspecția programului. Verificarea automată. Testarea sistemului. Testarea componentelor. Asigurarea siguranței și securității	2	0
S7. Ingineria securității. Conceptele de securitate. Gestiunea riscurilor de securitate. Conceperea securității. Vulnerabilitatea sistemului.	2	1
S8. Ingineria produselor program orientată pe servicii și pe aspecte. Serviciile – componente reutilizabile. Ingineria serviciilor. Dezvoltarea programelor	1	0
Total lucrări de laborator/seminare:	30/15	10/4

8. Referințe bibliografice

Principale	<ol style="list-style-type: none"> Ian Sommerville Software Engineering, Eight Edition. Addison-Wesley 2007. Ian Sommerville Software Engineering, Seventh Edition. Addison-Wesley 2004. Ian Sommerville Software Engineering, Sixth Edition. Addison-Wesley 2001.
Suplimentare	<ol style="list-style-type: none"> Craig Larman, Applying UML and Patterns, Third Edition. Addison Wesley, October 2004. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Ph.D., Jim Conallen, Kelli A. Houston. Object-Oriented Analysis and Design with Applications, Third Edition. Addison-Wesley, April 30, 2007.

9. Evaluare

Forma de învățământ	Periodică		Curentă	Proiect/teza de an	Examen final
	Atestarea 1	Atestarea 2			
Cu frecvență	10%	10%	10%	30%	40%
Cu frecvență redusă	20%			30%	50%
Standard minim de performanță					
Prezența și activitatea la prelegeri și lucrări de laborator Obținerea notei minime de „5” la fiecare dintre evaluări și lucrări de laborator					