



FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Națională de Știință și Tehnologie Politehnica București
1.2 Facultatea	Electronică, Telecomunicații și Tehnologia Informației
1.3 Departamentul	Electronică Aplicată și Ingineria Informației
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Specializarea	Ingineria Informației

2. Date despre disciplină

2.1 Denumirea disciplinei (ro)		Inginerie software					
(en)		Software Engineering					
2.2 Titularul activităților de curs			S.I./Lect. Dr. Elena Cristina STOICA				
2.3 Titularul activităților de seminar / laborator			S.I./Lect. Dr. Elena Cristina STOICA				
2.4 Anul de studiu	4	2.5 Semestrul	I	2.6. Tipul de evaluare	V	2.7 Regimul disciplinei	Ob
2.8 Tipul disciplinei	D	2.9 Codul disciplinei	04.D.07.O.009	2.10 Tipul de notare	Nota		

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	3	Din care: 3.2 curs	2.00	3.3 seminar/laborator	1
3.4 Total ore din planul de învățământ	42.00	Din care: 3.5 curs	28	3.6 seminar/laborator	14
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					27
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					
Pregătire seminarii/ laboratoare/proiecte, teme, referate, portofolii și eseuri					
Tutorat					0
Examinări					6
Alte activități (dacă există):					0
3.7 Total ore studiu individual	33.00				
3.8 Total ore pe semestru	75				
3.9 Numărul de credite	3				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea calculatoarelor, Structuri de date si algoritmi, Programare obiect-orientata
4.2 de rezultate ale învățării	Cunoștințe de algoritmi si structuri de date, programare într-un limbaj uzual

5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

5.1 Curs	Acces Internet si platforma Moodle, sala de curs cu proiector
----------	---



5.2 Seminar/ Laborator/Proiect	Sala de calculatoare, rețea de comunicație. Prezența obligatorie la laboratoare (conform regulamentului studiilor universitare de masterat în UPB).
-----------------------------------	---

6. Obiectiv general (*Se referă la intențiile profesorilor pentru studenți, la ceea ce studenții vor fi învățați în timpul cursului. Oferă o orientare cu privire la locul cursului în cadrul domeniului științific abordat, precum și la rolul pe care acesta îl are în cadrul specializării studiate. Vor fi descrise de o manieră generală tematicile abordate, justificarea includerii cursului în planul de învățământ al specializării studiate etc.*)

Studiul metodelor ingineresti de rezolvare de probleme reale prin elaborarea de produse software ale aplicatiei dedicate, cu metode, procedee si alte mijloace software organizate puse la dispozitie de experienta de rezolvare de probleme cu software de dezvoltare.

7. Competențe (*Capacitatea dovedită de a utiliza cunoștințe, aptitudini și abilități personale, sociale și/sau metodologice în situații de muncă sau de studiu și pentru dezvoltarea profesională și personală. Reflectă cerințele angajatorilor.*)

Specifice	Aplicarea cunoștințelor, conceptelor și metodelor elementare privitoare la metodologiile și instrumentele de Inginerie Software. Dezvoltarea sistemelor software complexe: pe baza unor metodologii și instrumentele de Inginerie Software. Proiectarea și utilizarea sistemelor de calcul și a rețelelor de calculatoare
Transversale (generale)	Capacitatea de a asigura planificarea și managementul proiectelor din domeniul ingineriei informației. Capacitatea de a lua decizii în vederea rezolvării problemelor curente, sau imprevizibile, care apar în procesul de exploatare a sistemelor de calcul. Capacitatea de a se informa și documenta permanent pentru dezvoltarea personală și profesională prin citirea literaturii de specialitate. Capacitatea de a comunica și de a prezenta conținut tehnic atât în limba română, cât și în limba engleză. Flexibilitate în utilizarea de noi sisteme și tehnologii în cadrul unei echipe în care membrii împreună ating un obiectiv bine definit, asumând în același timp roluri sau sarcini diferite. Capacitatea de a lua decizii în vederea rezolvării problemelor curente

8. Rezultatele învățării (*Sunt enunțuri sintetice referitoare la ceea ce un student va fi capabil să facă sau să demonstreze la finalizarea unui curs. Rezultatele învățării reflectă realizările studentului și mai puțin intențiile profesorului. Rezultatele învățării informează studenții despre ceea ce se așteaptă de la ei din punct de vedere al performanței, pentru a obține notele și creditele dorite. Sunt definite în termeni concreți, folosind verbe similare exemplurilor de mai jos și indică ceea ce se va urmări prin evaluare. Rezultatele învățării vor fi astfel redactate încât să fie evidențiată clar relația față de competențele definite la punctul 7.)*

Cunoștințe	Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau faptice. Enumeră cele mai importante etape care au marcat dezvoltarea domeniului Ingineriei Software. Definește noțiuni specifice domeniului de Inginerie Software. Describe noțiuni de bază și organizarea etapelor unui proiect software Cunoștințele acumulate în cadrul orelor de curs și laborator pot fi folosite în proiectarea și implementarea proiectelor software
-------------------	---



Aptitudini	<p><i>Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente).</i></p> <p>Selectează și grupează informații relevante din domeniul ingineriei software. Utilizează argumentat principii specifice în vederea proiectării și implementării proiectelor software folosind metodologii specifice ingineriei software Lucrează productiv în echipă. Verifică experimental soluții identificate. Rezolvă aplicații practice. Analizează și compară mai multe modele de date existente în domeniul ingineriei software Identifică soluții și elaborează planuri de proiectare a aplicațiilor software. Formulează concluzii la implementările realizate. Argumentează soluțiile identificate precum și modurile de rezolvare a problemelor de proiectare și implementare a aplicațiilor software.</p>
Responsabilitate și autonomie	<p><i>Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale.</i></p> <p>Selectează surse bibliografice potrivite și le analizează. Respectă principiile de etică academică, citând corect sursele bibliografice utilizate. Demonstrează receptivitate pentru contexte noi de învățare. Manifestă colaborare cu ceilalți colegi și cadre didactice în desfășurarea activităților didactice Demonstrează autonomie în organizarea situației/contextului de învățare sau a situației problemă de rezolvat Manifestă responsabilitate socială prin implicarea activă în viața socială studentescă/implicare în evenimentele din comunitatea academică Promovează/contribuie prin soluții noi, aferente domeniului de specialitate pentru a îmbunătăți calitatea vieții sociale. Conștientizează valoarea contribuției sale în domeniul ingineriei la identificarea de soluții viabile/sustenabile care să rezolve probleme din viața socială și economică (responsabilitate socială). Aplică principii de etică/deontologie profesională în analiza impactului tehnologic al soluțiilor propuse în domeniul de specialitate asupra mediului înconjurător. Analizează și valorifică oportunități de afaceri/de dezvoltare antreprenorială în domeniul de specialitate. Demonstrează abilități de management al situațiilor din viața reală (gestionarea timpului colaborare vs. conflict).</p>

9. Metode de predare (Se vor avea în vedere metode care să asigure predarea centrată pe student. Se va descrie modul în care se asigură participarea studenților la stabilirea propriului parcurs de învățare, cum se identifică eventualele rămăneri în urmă și ce măsuri remediale se adoptă în astfel de cazuri.)

Prezentarea prelegerilor de curs se face în amfiteatru cu facilități multimedia sau online în platforma Teams. Materialele de curs sunt: notele și prezentările de curs. Toate materialele sunt disponibile în format electronic pe platforma „Moodle” a facultății.

10. Conținuturi

CURS		
Capitolul	Conținutul	Nr. ore



1	1. Necesitatea ingineriei software, istoric, definiții. Proiectarea inginerescă, ciclul de proiectare, specificitatea proiectării software. Cicluri de viață al software-ului, tipuri de abordări în rezolvarea de probleme.	4
2	2. Analiza de sistem, definirea cerințelor, niveluri formale de specificare a cerințelor, documente cu cerințe, evoluție cerințe, etape ale fazei de specificare, structura documentului de definire a cerințelor și de specificare a cerințelor. Modelarea de sistem, context de sistem, analiza punctului de vedere, descriere de model sistem software. Modelare obiectuală	6
3	3. Proiectare software. Proiectare structurată: definiții, teorema structurii, componente elementare, structuri de control de bază. Tehnici de descriere a programelor. Rafinarea programelor. Proiectarea funcțională: diagrame de flux de date și diagrame de structură, dicționar de date, sinteza diagramelor de structură.	6
4	4. Modelare obiectuală. UML Limbaj unificat de modelare. Organizare a activității în echipa software. Stil de programare.	4
5	5. Implementare, verificare și validare de programe. Calitate a proiectării software. Metrice software. Software reutilizabil	4
6	6. Verificare finală	4
	Total:	28

Bibliografie:

(1) A. Sommerville – “Software engineering”, ed. VIII, Addison Wesley, 2007

LABORATOR

Nr. crt.	Conținutul	Nr. ore
1	Diagrame UML de clase	2
2	Diagrame UML de tip activitate	2
3	Diagrame UML de stare	2
4	Diagrame UML de interacțiune	2
5	Diagrame UML „use case”	2
6	Planificare. Utilitarul Microsoft Project	2
7	Colocviu	2
	Total:	14

Bibliografie:

(1) A. Sommerville – “Software engineering”, ed. VIII, Addison Wesley, 2007

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare	11.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------



11.4 Curs	- cunoașterea noțiunilor teoretice fundamentale; - cunoașterea modului de aplicare a teoriei la probleme specifice; - analiza diferențială a instrumentelor și metodelor teoretice.	Test scris de verificare și subiectele, acoperă întreaga materie, realizând o sinteză între parcurgerea teoretică comparativă a materiei și explicitarea prin exerciții și probleme a modelelor de aplicație.	60%
11.5 Seminar/laborator/proiect	- cunoașterea tipurilor de metodologii și utilizarea acestora în rezolvarea unei probleme date; - cunoașterea modului de utilizare a unor unelte pentru dezvoltarea, de aplicații software	Colocviu final de laborator, cuprinzând o componentă teoretică și o componentă practică. Componenta teoretică este verificată prin test; componenta practică este evaluată prin verificarea modului de rezolvare (implementare, testare, funcționare) de către student a unei probleme practice.	40%
11.6 Condiții de promovare			
- realizarea obiectivului cursului, dovedită prin obținerea a cel puțin 50% din punctaj			

12. Coroborarea conținutului disciplinei cu așteptările reprezentanților angajatorilor și asociațiilor profesionale reprezentative din domeniul aferent programului, precum și cu stadiul actual al cunoașterii în domeniul științific abordat și practicile în instituții de învățământ superior din Spațiul European al Învățământului Superior (SEİS)

Industria are o cerere importantă de ingineri calificați, cu specializări legate de proiectarea și întreținerea rețelelor de calculatoare și a aplicațiilor de sistem pentru sisteme distribuite, inclusiv pentru aplicații în Internet – cloud computing, pentru care este necesară cunoașterea detaliată a structurii pe niveluri, a protocoalelor și a serviciilor rețelelor de calculatoare și a rețelelor de calculatoare interconectate; dezvoltarea actuală de sisteme dedicate complexe, inclusiv echipamente de comunicații sau mobile, include obligatoriu cunoașterea problemelor și a soluțiilor acestora în cadrul rețelelor de calculatoare. Programa cursului răspunde concret acestor cerințe actuale de dezvoltare și evoluție, subscrise planului EUROPA 2020, a serviciilor din domeniul Inginerie Electronică. În contextul progresului tehnologic actual al tehnologiei informației, domeniile de activitate vizate cuprind toate aspectele vieții economice și sociale în care se urmărește trecerea spre societatea bazată pe cunoaștere, în care experiența umană este externalizată în sisteme independente, dedicate unor domenii diverse, de la sisteme utilitare curente, la domeniul telecomunicații, domeniul militar, domeniul de securitate (sisteme de supraveghere), domeniul automatizărilor industriale (sisteme de inspecție produse), robotică (sisteme de interfațare om-mașină) și altele.

Se asigură astfel absolvenților competențe adecvate cu necesitățile calificărilor actuale și o pregătire științifică și tehnică moderne, de calitate și competitive, care să le permită angajarea rapidă după absolvire, fiind perfect încadrat în politica Universității Politehnica din București, atât din punctul de vedere al conținutului și structurii, cât și din punctul de vedere al aptitudinilor și deschiderii internaționale oferite studenților.



Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și
Tehnologia Informației



25.10.2024

S.I./Lect. Dr. Elena Cristina
STOICA

S.I./Lect. Dr. Elena Cristina
STOICA

Data avizării în departament

Director de departament

Conf. Dr. Bogdan Cristian FLOREA

Data aprobării în Consiliul
Facultății

Decan

01.11.2024

Prof. Dr. Mihnea Udrea