



FIȘA DISCIPLINEI
ANUL UNIVERSITAR 2025 – 2026

1. Date despre program

1.1. Instituția de învățământ superior	UNIVERSITATEA DIN CRAIOVA
1.2. Facultatea	Automatică, Calculatoare și Electronică
1.3. Departamentul	Calculatoare și Tehnologia Informației
1.4. Domeniul de studii	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Forma de organizare	IF
1.7. Programul de studii	Calculatoare / L2060101010

2. Date despre disciplină

2.1. Denumirea disciplinei	Inginerie software						
2.2. Titularul activităților de curs	Prof.univ.dr.ing. Mihai L. MOCANU						
2.3. Titularul activităților de seminar/ laborator	Ș.l.dr.ing. Paul Stefan POPESCU						
2.4. Anul de studiu	III	2.5. Semestrul	II	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	DOB

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

3.1. Numărul de ore pe săptămână	5	din care: 3.2 curs	3	3.3. sem./lab./pr.	2
3.4. Total ore din planul de învățământ	70	din care: 3.5 curs	42	3.6. sem./lab./pr.	28
Distribuția fondului de timp - ore/sapt.					30/14
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					5
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					5
Tutoriat					-
Examinări					6
Alte activități.....					-
3.7. Total ore studiu individual					30
3.8. Total ore pe semestru					100
3.9. Numărul de credite					4

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Sunt necesare cunoștințe de bază, dobândite în cadrul cursurilor anterioare din anul I, de: • Programarea calculatoarelor • Tehnici de programare • Algoritmi și structuri de date • Programarea orientată pe obiecte • Proiectare orientată pe obiecte • Grafica pe calculator • Medii de programare vizuală	•
4.2. de competențe	Nu este cazul.	•

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Cursul se desfășoară în sălile de curs ale facultății, dotate cu videoproector. Există un manual de baza al cursului (textbook) utilizat pe scara largă pe plan mondial –Ian Sommerville, Software Engineering, ed.7 [1] și separat, prezentările actualizate anual (în format PowerPoint). La sfârșitul fiecărui curs are loc discuția noțiunilor predate. Lunar, sunt date teste de evaluare urmate de discuția lor, dar și teme de casă, de regulă la sfârșit de capitol, cu probleme ce fixează noțiunile prezentate.	•
5.2. de desfășurare a seminarului/ laboratorului	Activitățile aplicative se desfășoară în laboratoarele facultății - sala S1, dotată cu 10 calculatoare și sala S2 dotată cu 15 calculatoare. Instrumentele de lucru folosite sunt: Java, Eclipse, ArgoUML.	•

6. Obiectivele disciplinei - rezultate așteptate ale învățării la formarea cărora contribuie parcurgerea și promovarea disciplinei

Cunoștințe	Studentul/Absolventul: 1. descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare, baze de date. inteligență artificială și inginerie software și modul lor de aplicare în probleme concrete.
Aptitudini (Abilități)	Studentul/Absolventul: 1. alege și explică concepte proprii specifice proiectării algoritmilor, programării orientate pe obiecte, programării logice și funcționale. 2. specifică cerințe, analizează, elaborează, dezvoltă și testează sisteme software în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), aplicând elementele specifice ingineriei software.
Responsabilitate și autonomie	Studentul/Absolventul: 1. are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

7. Conținuturi

7.1. CURS	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
Curs 1: 1. 1.1. Fazele dezvoltării unui produs software (ciclul de viață program): 1.2. Metodologii de dezvoltare (proces de dezvoltare software): dezvoltarea în cascada, spirala, prototipizarea, metode formale, metodologii agile, programare extremă etc	Online sincron	Prezentări interactive, utilizând mijloace moderne de predare – calculator, software specializat de prezentare (PowerPoint, simularea mediului de programare), dar și sesiuni ilustrative de rezolvare probleme. Materialul predat la curs este grupat în capitole/ subcapitole, conform cu numerotarea din prima coloana.	3
Curs 2: 1.3. Managementul de proiect software: planificare și control; managementul configurației, managementul echipei. 1.4. Estimarea costului unui proiect software. 1.5. Reutilizarea resurselor	Online sincron		3
Curs 3: 2. Ingineria cerințelor: 2.1. Extragerea cerințelor. 2.2. Metode pentru identificarea cerințelor utilizatorilor. 2.3. Metode pentru specificarea cerințelor utilizatorilor. Documentul cerințelor	Online sincron		3
Curs 4: 2.4. Analiza orientata obiect.	Online sincron		3

Metode de analiza orientata obiect. 2.5. Modele de sistem. Modele contextuale. Modele comportamentale. Modele semantice de date. Modele obiectuale/ ierarhice. 2.6. Procese standard de obținere a modelelor de sistem			
Curs 5: 2.7. Modelare sistematică, vizuală în „limbajul de modelare unificator”. Vizibilitatea atributelor și operațiilor, interfețe, asociere, agregare, compunere, moștenire, polimorfism. Tipuri de diagrame UML: D. cazurilor de utilizare. D. de clase. Diagrame. de interacțiune: de secvență și de colaborare. D. de activități și de stare. D. de componente. D. de pachete. D. de implementare și de aranjare	Online sincron		3
Curs 6: 3. Proiectarea rațională orientată pe obiecte cu UML. 3.1. Procesul proiectării sistemelor software. 3.2. Caracteristici ale proiectării corecte: extensibilitatea, siguranța, eficiența sistemului	Online sincron		3
Curs 7: 3.3. Structuri abstracte în proiectarea software. Perspective arhitecturale ale sistemului. Arhitecturi de sistem distribuite. Arhitecturi de sistem orientate pe aplicații. 3.4. Etapele proiectării orientate obiect. Contextul sistemului și modelele de utilizare. Proiectarea arhitecturii. Identificarea obiectelor. Modele de proiectare. Specificarea interfețelor obiectelor	față în față		3
Curs 8: 3.5. Proiectarea interfețelor utilizator. Prototipizare. 3.6. Șabloane de proiectare	față în față		3
Curs 9: 4. Implementare și integrare pentru sistemele software. 4.1. Limbaje de programare. 4.2. Programarea structurată/ modulară. Caracteristici. Cuplarea modulelor. Principii euristice. Reutilizare	față în față		3
Curs 10: 4.3. Standarde de codificare. 4.4. Controlul versiunilor - implementare și integrare.	față în față		3
Curs 11: 5. Verificare și validare. 5.1. Siguranța în funcționare a sistemelor de programe. Prevenirea defectelor, planificarea, analiza statică, metode formale. Sisteme critice: redundanța statică și dinamică, restabilirea în toleranța la defectare, tratarea excepțiilor.	față în față		3
Curs 12: 5.2. Testarea: selecția cazurilor de test. Testarea modulelor: metode de testare: tip cutie neagră; structurale; inspecția codului. Teste de integrare. 5.3. Planificarea reviziilor	față în față		3

Curs 13: 6. Mentenanța și evoluția sistemelor software. 6.1. Managementul proceselor de mentenanță. 6.2. Planificarea versiunilor și noilor release-uri	față în față		3
Curs 14: 6.3. Ingineria inversă.	față în față		3

Bibliografie:

1. Ian Sommerville - *Software Engineering*, 9th Ed., Addison-Wesley 2010
2. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*, 2001, 2004, IEEE Press
3. A. Cockburn, *Surviving Object-Oriented Projects*, Addison-Wesley, 1998.
4. S. Roberson, and R. Robertson, *Managing Requirements*, Addison-Wesley, 1999
5. Kent Beck, *Extreme Programming Explained*, Addison-Wesley, 1999
6. Schach S.R., *Object-Oriented and Classical Software Engineering*, 6th Ed., McGraw Hill, 2006
7. Tom Pender - *UML Bible*, John Wiley & Sons, 2003

7.2. Seminar/laborator/proiect	Modalitatea de desfășurare	Metode de predare	Fond de timp alocat (ore)
Lab1 – Introducere. Analiza cerințelor. Șabloane	Online sincron	Implementare și prezentare aplicații pe tematica fiecărui laborator. Integrarea cunoștințelor într-o temă semestrială în pregătirea realizării proiectului de semestru la alegere dintr-o lista propusă la începutul de semestru. Discuția modalităților de raportare și prezentare ca și a modului de realizare a documentației proiectului.	2
Lab2 – Modelare cazuri de utilizare	Online sincron		2
Lab3 – Concepte OO. Modelare. Tipuri de modele.	Online sincron		2
Lab4 – Proiectare software I – Arhitectura.	față în față		2
Lab5 – Proiectare software II - Proiectare detaliată.	față în față		2
Lab6 – Modele de proiectare.	față în față		2
Lab7 – Diagrame UML pentru modelare funcțională.	față în față		2
Lab8 – Diagrame UML pentru modelare structurală.	față în față		2
Lab9 – Diagrame UML pentru modelare dinamică.	față în față		2
Lab10 – Șabloane de proiectare: creaționale, structurale și comportamentale.	față în față		2
Lab11 – Implementarea. Tratarea erorilor și excepțiilor	față în față		2
Lab12 – Controlul versiunilor.	față în față		2
Lab13 – Metrice Software. Testarea. Utilizarea programelor de testare	față în față		2
Lab14 – Testarea performanțelor. Benchmark-uri.	față în față		2

Bibliografie:

1. Pressman, Roger S (2009). *Software Engineering: A Practitioner's Approach* (7th ed.). Boston, Mass: McGraw-Hill. ISBN 978-0073375977.
2. Sommerville, Ian (2010) . *Software Engineering* (9th ed.). Harlow, England: Pearson Education. ISBN 978-0137035151.
3. Sinan Si Alhir - *Learning UML*, O'Reilly, 2003.
4. Joseph Schmuller - *Teach Yourself UML in 24 Hours*, Sams Publishing, 2004.
5. ArgoUML, <http://argouml.tigris.org/>.

8. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Continutul cursului este în acord cu recomandările reprezentanților comunității științifice, institutelor de cercetare internaționale în domeniu (PMI, SEI), asociațiilor profesionale internaționale (IEEE, ACM) și naționale (SRAIT) și conform cu cerințe exprimate de angajatorii reprezentativi (Hella, Ford, IQuest etc.).

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Înțelegerea noțiunilor teoretice fundamentale. Capacitatea de analiza și sinteza în situații concrete. Capacitatea de a realiza conexiuni între noțiunile prezentate.	Examen scris si/ sau oral Teste de tip grilă Teme de casă	40% 20% 20%
9.5.Seminar/laborator/proiect	L: Activitatea individuală dar și activitatea de lucru ca membru al unei echipe de proiect.	Nota laborator = $50\% * (\text{Nota1} + \text{Nota2}) + 50\% * \text{Nota3}$ Nota1: nota tema semestru (lucru în echipă) Nota2: nota la prezentarea individuala a părții proprii din tema semestrială Nota3: nota la activitatea individuala în clasă	20% Calculul notei finale se face prin ponderarea și rotunjirea punctajului final la o valoare întreagă între 1..10.
9.6. Standard minim de performanță Studentul trebuie să obțină cel puțin 50% (minim 10 puncte) la evaluările intermediare (de tip test grila, teme de casa sau laborator) și sa acumuleze apoi cel puțin 50% din punctajul aferent lucrării scrise (minim 20 puncte) pentru a promova.			

Data completării
01.10.2025

Titular de disciplină,
Prof.univ.dr.ing. Mihai L. MOCANU

Semnătura titularului

.....

Data avizării în departament
03.10.2025

Director de departament,
Ș.l.dr.ing. Nicolae ENESCU

Semnătura directorului de departament,

.....

Decan,
Prof.dr.ing. Dorin ȘENDRESCU

Semnătura decanului

.....