

# FIȘA DISCIPLINEI

## 1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Inginerie Mecanică și Electrică
1.3. Departamentul	Automatică, Calculatoare și Electronică
1.4. Domeniul de studii universitare	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Calculatoare

## 2. Date despre disciplină

2.1. Denumirea disciplinei	Programare logică și programare funcțională
2.2. Titularul activităților de curs	Prof. dr. ing. Mihaela Oprea
2.3. Titularul activităților seminar/laborator	Șef lucrări dr. mat. Mădălina Cărbureanu
2.4. Titularul activității proiect	
2.5. Anul de studiu	II
2.6. Semestrul *	4
2.7. Tipul de evaluare	Verificare
2.8. Categoria formativă** / regimul*** disciplinei	DD / DOB

\* numărul semestrului este conform planului de învățământ;

\*\* DF - Discipline fundamentale; DS - discipline de specializare; DC - discipline complementare

\*\*\* obligatorie/impusă = DOB; opțională = DOP; facultativă = DFA

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

3.1. Număr de ore pe săptămână	4	din care: 3.2. curs	2	3.3. Seminar/laborator	2	3.4. Proiect	
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. Seminar/laborator	28	3.8. Proiect	
3.9. Total ore studiu individual (studiu după suport de curs, bibliografie și notițe, documentare suplimentară în bibliotecă, pe platformele electronice de specialitate, pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri)							44
3.10. Total ore pe semestru							100
3.11. Numărul de credite							4

## 4. Condiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none"> <li>➤ Programarea calculatoarelor și limbaje de programare</li> <li>➤ Programare orientată pe obiecte</li> </ul>
4.2. de desfășurare a cursului	➤ În funcție de modalitatea de susținere a orelor de curs (față în față/online), sală de curs/Google Meet și platforma e-learning a facultății.
4.3. de desfășurare a seminarului/laboratorului	➤ Laborator dotat cu tehnică de calcul și medii de programare Prolog și Haskell (în cazul orelor susținute față în față). Platforma e-learning a facultății și Google Meet (în cazul orelor susținute online).

## 5. Competențe specifice acumulate și rezultatele învățării\* care stau la baza acestora

Competențe profesionale	Rezultatele învățării*
1. Operarea cu fundamente științifice, ingineresti și ale tehnologiei informației.	<p><b>C1</b> - Studentul/absolventul identifică și descrie concepte, principii și metode de bază din programarea declarativă (programarea logică și programarea funcțională).</p> <p><b>A1</b> - Studentul/absolventul analizează sistemele utilizând teoriile studiate și proiectează, implementează, diagnostichează și depanează programe logice și programe funcționale.</p>

	<b>RA1</b> - Studentul/absolventul selectează și utilizează surse bibliografice specifice domeniului.
2. Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor	<p><b>C1</b> - Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare declarativă (limbaje de programare logică și limbaje de programare funcțională), medii de programare, tehnici de programare și inginerie software specifică paradigmei de programare declarativă și modul lor de aplicare în probleme concrete.</p> <p><b>A1</b> - Studentul/absolventul alege și explică concepte proprii specifice programării declarative (programarea logică și programarea funcțională).</p> <p><b>RA1</b> - Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.</p> <p><b>RA2</b> - Studentul/absolventul selectează și utilizează surse bibliografice specifice domeniului.</p>
3. Proiectarea, gestionarea ciclului de viață și integrarea sistemelor informatice utilizând tehnologii și medii de programare	<p><b>C1</b> - Studentul/absolventul identifică, descrie și sumarizează concepte și metode elementare privitoare la limbaje de programare declarativă (limbaje de programare logică – Prolog, limbaje de programare funcțională – Haskell), medii de programare, tehnici de programare și inginerie software specifică paradigmei de programare declarativă și modul lor de aplicare.</p> <p><b>A1</b> - Studentul/absolventul elaborează specificații și proiectează sisteme informatice folosind metode și instrumente specifice programării logice și programării funcționale.</p> <p><b>RA1</b> - Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.</p> <p><b>RA2</b> - Studentul/absolventul selectează și utilizează surse bibliografice specifice domeniului.</p>
<b>Competențe transversale</b>	<b>Rezultatele învățării*</b>
1. Aplicarea, în mod responsabil, a principiilor, normelor și valorilor eticii profesionale în realizarea sarcinilor profesionale și identificarea obiectivelor de realizat, a resurselor disponibile, a etapelor de lucru, a duratelor de execuție, a termenelor de realizare aferente și a riscurilor aferente.	<p><b>C1</b> - Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale din programarea declarativă (programarea logică și programarea funcțională) și modul lor de aplicare în probleme concrete.</p> <p><b>A1</b> - Studentul/absolventul specifică cerințe, elaborează scenarii de simulare, propune soluții de rezolvare a unor probleme de control (folosind programarea declarativă – programarea logică și programarea funcțională), analizează și evaluează performanțele sistemelor informatice bazate pe paradigma programării declarative.</p> <p><b>RA1</b> - Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.</p> <p><b>RA2</b> - Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer.</p>
2. Identificarea oportunităților de formare continuă și utilizarea eficientă pentru propria dezvoltare a surselor informaționale și a resurselor de comunicare și formare profesională asistată de calculator (portaluri Internet, aplicații software de specialitate, baze de date, cursuri on-line) atât în limba română, cât și într-o limbă de circulație internațională.	<p><b>C1</b> - Studentul/absolventul explică și interpretează rezultate teoretice și experimentale, documentație tehnică, fenomene și procese din domeniul calculatoare și tehnologia informației – programarea declarativă (programarea logică și programarea funcțională).</p> <p><b>A1</b> - Studentul/absolventul realizează responsabil proiecte pentru rezolvarea unor probleme specifice domeniului, cu evaluarea corectă a volumului de lucru, a resurselor disponibile, a timpului necesar de finalizare și a riscurilor, în condiții de aplicare a normelor deontologice și de etică profesională în domeniu, precum și de securitate și sănătate în muncă.</p> <p><b>RA1</b> - Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.</p> <p><b>RA3</b> - Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate.</p>

\* C – cunoștințe; A – aptitudini; RA – responsabilitate și autonomie.

## 6. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

6.1. Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>➤ La sfârșitul cursului, studentul va fi capabil să dezvolte aplicații software, folosind două stiluri de programare ale paradigmei declarative: programarea logică și programarea funcțională.</li> </ul>
6.2. Obiectivele specifice	<ul style="list-style-type: none"> <li>➤ La sfârșitul cursului, studentul va fi capabil</li> <li>➤ să definească conceptele fundamentale ale programării logice și ale programării funcționale;</li> <li>➤ să explice conceptele fundamentale ale programării logice și ale programării funcționale;</li> </ul>

	➤ să aplice conceptele fundamentale ale programării logice în limbajul Prolog și ale programării funcționale în limbajul Haskell.
--	---

## 7. Conținuturi

7.1. Curs	Nr. ore	Metode de predare	Observații
Paradigme ale programării calculatoarelor	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Introducere în programarea logică	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Limbajul de programare logică Prolog - Prezentare generală și elemente de bază	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Recursivitate în limbajul Prolog. Liste Prolog	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Fișiere și baze de date în limbajul Prolog. Alte facilități ale limbajului Prolog. Aplicații ale limbajului Prolog	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Logica cu predicate de ordinul I	4	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Introducere în programarea funcțională	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Limbajul Haskell – Prezentare generală. Elemente de bază	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Recursivitate în limbajul Haskell. Liste Haskell	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Evaluarea leneșă. Monade. Alte facilități ale limbajului Haskell. Aplicații ale limbajului Haskell	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)
Calculul Lambda	2	interactivă și convențională, centrată pe student	Suport de curs în format tipărit și electronic (slide-uri)

Aplicații	4	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	Suport de curs în format tipărit și electronic (slide-uri)
<b>Bibliografie</b> [1] M. Oprea (2020) <i>Programare logică și Programare funcțională – Teorie și Aplicații</i> , Editura Matrix Rom, București. [2] M. Bramer (2013) <i>Logic programming with Prolog</i> , Springer. [3] M. Oprea (1999) <i>Inteligență artificială – Programare logică și Sisteme expert</i> , vol. II, UPG Ploiești. [4] R. Bird (1998) <i>Introduction to Functional Programming</i> , 2nd edition, Prentice Hall, New York. [5] C. Giumale (1997) <i>Programare funcțională - o perspectivă pragmatică</i> , Editura Tehnică, București.			
<b>7.2. Seminar / laborator</b>	Nr. ore	Metode de predare	Observații
Recapitulare programare procedurală (limbajele C/C++). Paradigme ale programării calculatoarelor	4	clasică, centrată pe student și pe rezultatele învățării, dezbateri studii de caz	
Introducere în programarea logică. Logica cu predicate de ordinul I. Exerciții	2	clasică, centrată pe student și pe rezultatele învățării	îndrumar de laborator
Limbajul de programare logică Prolog - Prezentare generală și elemente de bază. Aplicații	2	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Recursivitate în limbajul Prolog. Liste Prolog. Aplicații	2	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Fișiere și baze de date în limbajul Prolog. Alte facilități ale limbajului Prolog. Aplicații	4	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Sisteme expert în limbajul Prolog. Aplicații	2	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Introducere în programarea funcțională. Calculul lambda	2	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Limbajul Haskell – Prezentare generală. Elemente de bază. Aplicații	2	clasică, centrată pe student și pe rezultatele învățării	îndrumar de laborator
Recursivitate în limbajul Haskell. Liste Haskell. Aplicații	4	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Evaluarea leneșă. Monade. Alte facilități ale limbajului Haskell. Aplicații	4	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
<b>Bibliografie</b> [1] M. Oprea (2020) <i>Programare logică și Programare funcțională – Teorie și Aplicații</i> , Editura Matrix Rom, București. [2] M. Oprea (2009) <i>Inteligență artificială - îndrumar de laborator</i> , Editura UPG Ploiești. [3] M. Oprea (1999) <i>Inteligență artificială – Programare logică și Sisteme expert</i> , vol. II, UPG Ploiești. [4] D. Medak, G. Navratil (2003) <i>Haskell Tutorial</i> , Technical University, Vienna.			
<b>7.3. Proiect</b>	Nr. ore	Metode de predare	Observații

Bibliografie

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

➤ Conținuturile disciplinei sunt specifice domeniului dezvoltării software-ului și sunt coroborate cu așteptările comunității epistemice, a asociațiilor profesionale și angajatorilor ce activează în acest domeniu.

**9. Evaluare**

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Verificare finală	Lucrare scrisă cu subiecte teoretice și aplicații	50%
	Teste la curs	Lucrare scrisă, examinare orală	10%
9.5. Seminar/laborator	Activitate laborator și verificări periodice	Examinare orală și program la calculator	30%
	Verificări pe parcurs	Examinare orală, lucrare scrisă și program la calculator	10%
9.6. Proiect			
9.7. Standard minim de performanță			
➤ Cunoașterea conceptelor fundamentale ale programării logice, la nivel teoretic și practic (programare în limbajul Prolog).			
➤ Cunoașterea conceptelor fundamentale ale programării funcționale, la nivel teoretic și practic (programare în limbajul Haskell).			

Data completării      Semnătura titularului de curs      Semnătura titularului de seminar/laborator      Semnătura titularului de proiect

17.09.2025      \_\_\_\_\_      \_\_\_\_\_      \_\_\_\_\_

Data avizării în departament      Director de departament      Decan  
26.09.2025      Conf. dr. ing. Pricop Emil      Conf. dr. ing. Bădicioiu Marius

\_\_\_\_\_