

FIȘA DISCIPLINEI¹⁾

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	III
2.6. Semestrul *	6
2.7. Tipul de evaluare	Verificare
2.8. Categoria formativă** / regimul*** disciplinei	DD / O

* numărul semestrului este conform planului de învățământ;

** DF - Discipline fundamentale; DD - discipline de domeniu; DS - discipline de specialitate; DC - discipline complementare, DA - disciplina de aprofundare, DSI- disciplina de sinteza.

*** obligatorie = O; opțională = A; facultativă = L

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. Distribuția fondului de timp							ore
Studiu după manual, suport de curs, bibliografie și notițe							12
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							12
Tutoriat							4
Examinări							4
Alte activități							2
3.10 Total ore studiu individual	44						
3.11. Total ore pe semestru	100						
3.12. Numărul de credite	4						

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none">➤ Programarea calculatoarelor și limbaje de programare➤ Programare orientată pe obiecte
--------------------	--

¹⁾ Adaptare după Ordinul Ministrului educației, cercetării, tineretului și sportului nr. 5 703/2011 privind implementarea Codului național al calificărilor din învățământul superior, publicat în Monitorul Oficial al României, partea I, nr.880 bis / 13.XII.2011

4.2. de competențe	➤ Operarea cu fundamente ale informaticii specifice programării calculatoarelor într-un limbaj de programare.
--------------------	---

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	➤ În funcție de modalitatea de susținere a orelor de curs (față către față/online), sală de curs/Google Meet și platforma e-learning a facultății.
5.2. de desfășurare a seminarului/laboratorului	➤ Laborator dotat cu tehnică de calcul și medii de programare Prolog și Haskell (în cazul orelor susținute față către față). Platforma e-learning a facultății și Google Meet (în cazul orelor susținute online).

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> ➤ Cunoașterea și înțelegerea principalelor concepte și fundamente din domeniul informaticii aplicate (CpS-1) – concepte și fundamente ale programării declarative (programare logică și programare funcțională); ➤ Operarea cu metode și concepte moderne din știința calculatoarelor (CpS-3) – operarea cu metode și concepte specifice programării declarative (programare logică și programare funcțională); ➤ Cunoașterea și utilizarea limbajului de programare logică Prolog, respectiv a limbajului de programare funcțională Haskell, pentru aplicații practice (CpS-4); ➤ Proiectarea și implementarea aplicațiilor, utilizând principii de management de proiect (CpS-9) – dezvoltarea unor aplicații folosind două stiluri de programare specifice paradigmei declarative, programarea logică și programarea funcțională, în cadrul unei echipe de lucru.
Competențe transversale	<ul style="list-style-type: none"> ➤ Comportarea onorabilă, responsabilă, în spiritul eticii profesionale și respectării legislației curente (inclusiv a drepturilor de proprietate intelectuală), pentru a asigura reputația statutului de student și a profesiei alese (CtS-1); ➤ Demonstrarea spiritului de integrare, de inițiativă și de identificare a problemelor și responsabilităților din cadrul unei echipe de lucru interdisciplinare și plurispecializate (CtS-3); ➤ Abilitatea de a conduce eficient un colectiv de lucru, de a comunica în bune condiții, de a lua decizii în timp real, de a distribui sarcini și verifica îndeplinirea acestora la toate nivelurile subordonate (CtS-7).

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

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

8. Conținuturi

8.1. 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		
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)

Bibliografie

[1] M. Oprea (2020), *Programare logică și Programare funcțională – Teorie și Aplicații*, Editura Matrix Rom, București.

[2] M. Bramer (2013), *Logic programming with Prolog*, Springer.

[3] M. Oprea (1999), *Inteligență artificială – Programare logică și Sisteme expert*, vol. II, UPG Ploiești.

[4] R. Bird (1998), *Introduction to Functional Programming*, 2nd edition, Prentice Hall, New York.

[5] C. Giumale (1997), *Programare funcțională - o perspectivă pragmatică*, Editura Tehnică, București.

8.2. Seminar / laborator	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. Calcul 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
Bibliografie [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.			
8.3. Proiect	Nr. ore	Metode de predare	Observații
Bibliografie			

9. 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.

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Verificare finală	Lucrare scrisă cu subiecte teoretice și aplicații	50%
	Teste la curs	Lucrare scrisă, examinare orală	10%
10.5. 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%
10.6. Proiect			
10.7. 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
------------------	-------------------------------	--	----------------------------------

20.09.2021

Data avizării în
departament

28.09.2021

Director de departament
Conf. dr. ing. PRICOP Emil
(Semnătură)

Decan
Conf. dr. ing. DINIȚĂ Alin
(Semnătură)