



**FIȘA DISCIPLINEI**  
(licență)

**1. Date despre program**

1.1. Instituția de învățământ superior	UNIVERSITATEA „VASILE ALECSANDRI” din Bacău
1.2. Facultatea	DE INGINERIE
1.3. Departamentul	ENERGETICĂ ȘI ȘTIINȚA CALCULATOARELOR
1.4. Domeniul de studii	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.5. Ciclul de studii	LICENȚĂ
1.6. Programul de studii/calificarea	TEHNOLOGIA INFORMAȚIEI
1.7. Forma de învățământ	ÎNVĂȚĂMÂNT CU FRECVENȚĂ

**2. Date despre disciplină**

2.1. Denumirea disciplinei	Programare logică și programare funcțională				
2.2. Titularul activităților de curs	Lect.Dr Dan Popa				
2.3. Titularul activităților de seminar	Lect Dr. Dan Popa				
2.4. Anul de studiu	2	2.5. Semestrul	1	2.6. Tipul de evaluare examen	E
2.7. Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DD - în domeniu, <b>DS - de specialitate</b> , DC - complementară				DD
	Categorია de opționalitate a disciplinei: <b>DI - obligatorie (impusă)</b> , DO - opțională (la alegere), DL - facultativă (liber aleasă)				DI

**3. Timpul total estimat (ore alocate activităților didactice)**

3.1. Număr de ore pe săptămână	3	3.2. Curs	2	3.3. Seminar/Laborator/Proiect	1
3.4. Totalul de ore pe semestru din planul de învățământ	42	3.5. Curs	28	3.6. Seminar/Laborator/Proiect	14

Distribuția fondului de timp pe semestru:	ore
Studiul după manual, suport de curs, bibliografie și notițe	50
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	
Pregătire seminar/laboratoare, teme, referate, portofolii și eseuri	8
Tutoriat	
Examinări	
Alte activități (precizați):	

3.7. Total ore studiu individual	58
3.8. Total ore pe semestru	100
3.9. Numărul de credite	4

**1. Precondiții (acolo unde este cazul)**

4.1. de curriculum	<ul style="list-style-type: none"> <li>•Algebră, logică matematică, teoria funcțiilor.</li> </ul>
4.2. de competențe	<ul style="list-style-type: none"> <li>•Abilități de utilizare a S.O. Linux sau utilizare a S.O. Windows.</li> <li>•Editare de text, operare Linux, operarea unui mediu de programare.</li> </ul>

**2. Condiții (acolo unde este cazul)**

5.1. de desfășurare a cursului	<ul style="list-style-type: none"> <li>Sala de curs prevăzută cu tablă și videoproiector, convertor VGA to HDMI invers, după caz.</li> </ul>
5.2. de desfășurare a seminarului/laboratorului/proiectului	<ul style="list-style-type: none"> <li>Sala de laborator prevăzută cu computere cu S.O. Alternative (Linux) tablă, videoproiector, convertor VGA to HDMI sau invers, după caz.</li> </ul>

### 3. Competențe specifice acumulate

6.1. Competențe profesionale	C1 Operarea cu fundamente științifice, ingineresti și ale informaticii C4 Proiectarea și integrarea sistemelor informatice utilizând tehnologii și medii de programare
6.2. Competențe transversale	

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

7.1. Obiectivul general al disciplinei	C4.1. Definirea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice
7.2. Obiectivele specifice	Programarea logică bazată pe logica matematică și cea funcțională bazată pe $\lambda$ calcul

### 5. Conținuturi

Curs : Teoria $\lambda$ calculului	Nr. ore	Metode de predare	Observații
1. Introducere în lambda calcul: 1.1. Sintaxa și semantica $\lambda$ calculului; 1.2 Notății; 1.3 Variabile libere și legate; 2. 1.4 Reguli de conversie 1.4.1 Alfa conversia; 1.4.2 Beta conversia; 1.4.3 Eta conversia ; 1.5 Egalitatea lambda expresiilor; 1.6. Extensionalitate 3. 1.7 Substituție 4. Reprezentarea datelor în lambda calcul: 2.1 Valorile booleene și condiționala; 2.2 Perechi și n-uple; 5. 2.3 Numere; 2.4 Definiții prin recursie; 2.5 Funcții cu mai multe argumente; 2.6 Recursie mutuala; 2.7 6. Reprezentarea funcțiilor recursive; Combinatorul Y; 2.8 Extinderile $\lambda$ calculului; 7. 2.9 Rezultate teoretice despre $\lambda$ calculul, Teorema Church Roser, Corolarii ; Teorema de normalizare, 2.10 Folosirea lui Y în limbaje;e cu apeluri Call by Value . Introducere în teoria combinatorilor.	14 +	Prelegere orală, exemple de calcul	
<b>Teoria programării logice</b> 1. Logica predicatelor, logica propozițiilor, Prolog, SWI-PROLOG 2. Funcționarea Prolog-ului: Deducție, Demonstrație, Rezoluție, Unificare, Algoritmul lui Robinson, Backtracking. 3. Prologul ca limbaj de programare. Structuri de date: Liste și recursivitate. 4. Prologul ca limbaj de programare. Structuri de date: Arbori și recursivitate. 5. Controlul execuției in Prolog. Programare in stil imperativ in Prolog. Cut. Structurile programarii structurate și implementarea in prolog. 6. Metalogica sau Fold-uri. Fold-based programming. 7. Baze de date in Prolog. Accelerarea programelor	+ 14	Prelegere orală, exemple de programe	

Prolog.			
<b>Bibliografie</b>			
<ul style="list-style-type: none"> <li>• [1] A.L. Johanson, A.Eriksson-Granskog A. Edman – Prolog Versus You, Springer Verlag, 1989</li> <li>• [2] Gontineac, Mihai, <i>Programare Funcțională</i> O introducere utilizând limbajul Haskell - Ed. Alexandru Myller, Iași , 2006</li> <li>• [3] Popa, Dan V., <i>O introducere în Haskell 98 prin exemple</i>, ediția a doua republicată, Bacău: Editura Alma Mater, 2015, 312 pagini, ISBN 978-606-527-506-5.</li> <li>• [4]. Popa, Dan V., <i>Introducere în limbajul SWI-PROLOG</i>, Bacău: Editura Alma Mater, 2016, 138 pagini, ISBN 978-606-527-522-5.</li> <li>• [5] Hudak, Paul; Peterson John; Fasel Joseph; A Gentle Introduction to Haskell 98; online edition october, 1999</li> <li>• [6] Hudak, Paul; Peterson John; Fasel Joseph; (Dan Popa trad.) O mica introducere in Haskell 98; Ed. Matrixrom, București 2011, ISBN 978-973-755-701-8</li> <li>• [7] Dan Popa, Using Prolog for the study of algebraic structures and complex operations,</li> <li>• CREATIVE MATHS, pp 43-50 nr 12 (2003), Univ. de Nord Baia Mare</li> <li>• [8] DAN POPA, Funcțional fold-based programming in SWI-PROLOG, "Vasile Alecsandri" University of Bacău ,Faculty of Sciences, Scientific Studies and Research Series Mathematics and Informatics,Vol. 24 (2014) – pe site-ul pubs.ub.ro.</li> <li>• [9] <a href="http://www.swi-prolog.org">www.swi-prolog.org</a> – Site al proiectului SWI-Prolog</li> <li>• [10] <a href="http://www.haskell.org/haskellwiki/Ro/Haskell">www.haskell.org/haskellwiki/Ro/Haskell</a> – Site al comunitatii de Haskell</li> <li>• [11] Tudor, Nicoleta Liviana – Programare logică și sisteme expert Matrix Rom 2012</li> </ul>			
<b>Bibliografie minimală</b>			
<ul style="list-style-type: none"> <li>• [1] Popa, Dan V., <i>O introducere în Haskell 98 prin exemple</i>, ediția a doua republicată, Bacău: Editura Alma Mater, 2015, 312 pagini, ISBN 978-606-527-506-5.</li> <li>• [2]. Popa, Dan V., <i>Introducere în limbajul SWI-PROLOG</i>, Bacău: Editura Alma Mater, 2016, 138 pagini, ISBN 978-606-527-522-5.</li> <li>• [3] <a href="http://www.swi-prolog.org">www.swi-prolog.org</a> – Site al proiectului SWI-Prolog</li> </ul>			

Aplicații (Seminar / laborator / proiect)	Nr. ore	Metode de predare	Observații
1. Introducere în limbaje de programare funcționale	1		
2. Valori, tipuri și alte entități: 2.1. Tipuri polimorfe; 2.2. Tipuri definite de utilizator; 2.2.1. Tipuri recursive; 2.3. Sinonimie de tipuri; 2.4. Tipuri predefinite versus tipuri utilizator ; 2.4.1. Liste și secvențe aritmetice ; 2.4.2. String-uri	1		
3. Funcții: 3.1 Lambda – abstracții ; 3.2 Operatori infixaiți ; 3.2.1 Selecții ; 3.2.2 Declarații de prioritate pentru operatori ; 3.3 Funcțiile sunt nestricte; 3.4 Structuri de date ciclice, liste infinite ; 3.5 Funcția "error"	1		
4. Expresii Case și Șabloane (Pattern Matching) : 4.1 Semantica Șabloanelor ; 4.2. Un exemplu de utilizare ; 4.3 Expresii Case ; 4.4 Șabloane "leneșe" (cu evaluare întârziată) ; 4.5. Expresii "let" și clauze "where" – definirea de variabile locale ; 4.6. Layout-ul Polimorfism parametric, clase și supraîncărcarea operatorilor ;	1		
5. Clase, comparații cu alte limbaje Tipuri: 6.1 Declarația newtype ; 6.2 Câmpuri etichetate	1		
6. Input/Output: 7.1 Operații de I/O de bază; 7.2 Programarea cu acțiuni (do); 7.3 Manipularea excepțiilor; 7.4 Fișier, Canal, Handler; 7.5 Haskell și programarea imperativă	1		
7. Clase standard în Haskell: 8.1 Egalitatea și clasele (valorilor) ordonate; 8.2 Enumerări; 8.3 Clasele Read și Show; 8.4 Instanțe derivate	1		

Bibliografie	
<ul style="list-style-type: none"> <li>[1] A.L. Johanson, A.Eriksson-Granskog A. Edman – Prolog Versus You, Springer Verlag, 1989</li> <li>[2] Gontineac Mihai, <i>Programare Funcțională</i> O introducere utilizând limbajul Haskell - Ed. Alexandru Myller, Iași , 2006</li> <li>[3] Popa, Dan V., <i>O introducere în Haskell 98 prin exemple</i>, ediția a doua republicată, Bacău: Editura Alma Mater, 2015, 312 pagini, ISBN 978-606-527-506-5.</li> <li>[4] Popa, Dan, <i>Întrebări despre Haskell 98</i>, 2017, Publisher: AlmaMater, ISBN: 978-606-527-521-8 <a href="https://www.researchgate.net/publication/340581373_Intrebari_despre_Haskell_98">https://www.researchgate.net/publication/340581373_Intrebari_despre_Haskell_98</a></li> <li>[5] Popa, Dan V., <i>Introducere în limbajul SWI-PROLOG</i>, Bacău: Editura Alma Mater, 2016, 138 pagini, ISBN 978-606-527-522-5.</li> <li>[6] Hudak, Paul; Peterson John; Fasel Joseph; A Gentle Introduction to Haskell 98; online edition october, 1999</li> <li>[7] Hudak, Paul; Peterson John; Fasel Joseph; (Dan Popa trad.) O mică introducere in Haskell 98; Ed. Matrixrom, București 2011, ISBN 978-973-755-701-8</li> <li>[8] <a href="http://www.swi-prolog.org">www.swi-prolog.org</a> – Site al proiectului SWI-Prolog</li> <li>[9] Ulf Nilsson and Jan Małuszzyński LOGIC, PROGRAMMING AND PROLOG (2ED) <a href="https://www.ida.liu.se/~ulfni53/lpp/bok/bok.pdf">https://www.ida.liu.se/~ulfni53/lpp/bok/bok.pdf</a></li> <li>[10] Max Bramer, Logic Programming, with Prolog, Springer, 2005 <a href="https://drive.uqu.edu.sa/~fbshareef/files/textbook.pdf">https://drive.uqu.edu.sa/~fbshareef/files/textbook.pdf</a></li> <li>[11] <a href="http://www.haskell.org/haskellwiki/Ro/Haskell">www.haskell.org/haskellwiki/Ro/Haskell</a> – Site al comunității de Haskell</li> <li>[12] Tudor, Nicoleta Liviana – Programare logică și sisteme expert Matrix Rom 2012</li> <li>[13] Lipovaca, Miran, Learn You a Haskell for Great Good ! , No Starch Press, 2011 <a href="https://www.oreilly.com/library/view/learn-you-a/9781457100406/">https://www.oreilly.com/library/view/learn-you-a/9781457100406/</a></li> <li>[14] Vitaly Bragilevsky, Haskell in Depth, Manning Publicationms 2021 <a href="https://www.oreilly.com/library/view/haskell-in-depth/9781617295409/">https://www.oreilly.com/library/view/haskell-in-depth/9781617295409/</a> <a href="https://www.amazon.co.uk/dp/161729540X?linkCode=gs2&amp;tag=uuid07-21">https://www.amazon.co.uk/dp/161729540X?linkCode=gs2&amp;tag=uuid07-21</a></li> <li>[15] Alvez, Claudia, Haskell: The Ultimate Beginner's Guide to Learn Haskell Programming Step by Step, 2020</li> </ul>	
Bibliografie minimală	
<ul style="list-style-type: none"> <li>[1] <a href="http://wiki.haskell.org/haskellwiki/Ro/Haskell">wiki.haskell.org/haskellwiki/Ro/Haskell</a> – Site al comunității de Haskell, creat de Dan V. Popa</li> <li>[2] Popa, Dan V., <i>O introducere în Haskell 98 prin exemple</i>, ediția a doua republicată, Bacău: Editura Alma Mater, 2015, 312 pagini, ISBN 978-606-527-506-5.</li> <li>[3] Popa, Dan V., <i>Introducere în limbajul SWI-PROLOG</i>, Bacău: Editura Alma Mater, 2016, 138 pagini, ISBN 978-606-527-522-5.</li> <li>[4] Dan Popa, <i>Întrebări despre Haskell 98</i>, 2017, Publisher: AlmaMater, ISBN: 978-606-527-521-8 <a href="https://www.researchgate.net/publication/340581373_Intrebari_despre_Haskell_98">https://www.researchgate.net/publication/340581373_Intrebari_despre_Haskell_98</a></li> </ul>	

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

- Este coroborat fiind curs de nivel european de specialitate cu acces la o nișă de pe piața muncii: Programare în limbajele inteligenței artificiale.

## 7. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Evaluarea culturii generale (în domeniu) corespunde setului de întrebări nr. 1. Evaluarea cunoștințelor teoretice (în domeniu) corespunde setului de întrebări nr. 2.	Examinare orală.	50%
10.5. Seminar/laborator/proiect	Evaluarea abilităților de programator și rezolvitor de exerciții (în domeniu) corespunde setului de întrebări nr. 3. Evaluarea cunoștințelor suplimentare nivelului 3, individual acumulate corespunde setului de întrebări nr. 4.	Examinare prin întrebări care verifică existența abilităților practice și experienței ce decurge din ele și (la nivelul 4) integrarea teoriei cu practica.	50%

10.6. Standard minim de performanță:

- Nivelul 1 + Nivelul al 2-lea – a se vedea 10.4

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar
20.9.2021	<b>Lector dr. Dan Popa</b>	<b>Lector dr. Dan Popa</b>

Data avizării în departament	Semnătura directorului de departament
23.09.2021	Prof.dr.ing. George Culea

Data aprobării în Consiliul Facultății	Semnătura decanului
27.09.2021	Conf.dr.ing. Mirela Panainte-Lehăduș