FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5 Ciclul de studii	Licenţă
1.6 Programul de studii / Calificarea	Calculatoare si Tehnologia Informatiei/ Inginer
1.7 Forma de învăţământ	IF – învăţământ cu frecvenţă
1.8 Codul disciplinei	32.

2. Date despre disciplină

2.1 Denumirea disciplinei				***Programare funcţională***	
2.2 Titularii de curs				Conf.dr.ing. Radu Razvan Slavescu - Radu.Razvan.Slavescu@cs.utcluj.ro	
2.3 Titularul/Titularii activităţilor de seminar/laborator/proiect				Prof.dr.ing. Camelia Pintea ing. Adrian Nandrean ing. Loredana Coroama ing. Sara Popa ing. Vanessa Mercea ing. Sonia Cibu	
2.4 Anul de studiu	3	2.5 Semestrul	5	2.6 Tipul de evaluare *(E – examen, C – colocviu, V – verificare)*	E
2.7 Regimul disciplinei	*DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară*				DD
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	Curs	2	Seminar		Laborator	2	Proiect	
3.2 Număr de ore pe semestru	56	din care:	Curs	28	Seminar		Laborator	28	Proiect	
3.3 Distribuţia fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie şi notiţe										15
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate şi pe teren										14
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii şi eseuri										10
(d) Tutoriat										2
(e) Examinări										3
(f) Alte activităţi:										

3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))	44
3.5 Total ore pe semestru (3.2+3.4)	100
3.6 Numărul de credite	4

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	
4.2 de competenţe	Notiuni fundamentale de algoritmi. Recursivitate

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Tabla, proiector, calculator
5.2. de desfăşurare a laboratorului	Calculatoare, interpretoare/compilatoare specifice limbajelor studiate (LISP, ML, Haskell), Prezenţă obligatorie 100% pentru admiterea la examenul final

6. Competenţele specifice acumulate

6.1 Competenţe profesionale	**C2 -** Proiectarea componentelor hardware, software şi de comunicaţii **C2.1 -** Descrierea structurii şi funcţionării componentelor hardware, software şi de comunicaţii **C2.2 -** Explicarea rolului, interacţiunii şi funcţionării componentelor sistemelor hardware, software şi de comunicaţii **C2.3 -** Construirea unor componente hardware, software şi de comunicaţii

	folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale şi tehnologii **C2.4 -** Evaluarea caracteristicilor funcţionale şi nefuncţionale ale componentelor hardware, software şi de comunicaţii, pe baza unor metrici **C2.5 -** Implementarea componentelor hardware, software şi de comunicaţie
6.2 Competenţe transversale	N/A

7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Obiectivul general este cunoasterea de către studenti a elementelor de baza ale paradigmei functionale de programare, a avantajelor acestei paradigme (imutabilitatea variabilelor, posibilitatea demostrarii formale a corectitudiniii unui program, posibilitatea paralelizarii facile a codului), precum și a fundamentelor sale teoretice (calcul lambda)
7.2 Obiectivele specifice	Pentru atingerea acestor obiective generale, studenţii vor invata: - sa scrie cod în maniera functionala, cu eliminarea variabilelor de stare - sa identifice avantajele si dezavantajele diferitelor stiluri de programare - sa utilizeze recursivitatea și sa structureze codul pentru optimizarea sa - sa demonstreze formal corectitudinea unui program - sa manipuleze expresii lambda de baza

8. Conţinuturi

8.1 Curs	Nr.ore	Metode de predare	Observaţii
Noţiuni fundamentale de programare în Haskell şi ML: funcţii, constante, identificatori.	2	(Fizic/online) Slideuri, Demonstratii si reprezentare modele pe tabla, exercitii rapide pentru cresterea interactiunii	
Noţiuni fundamentale: tipuri primitive de date, recursivitate, liste, tupluri, operatori infix, evaluare.	2		
Liste: construire listă, operaţii fundamentale pe liste. Noţiuni fundamentale: declaraţii locale, tipuri polimorfice.	2		
Liste: operatori de liste (generatori, filtre, expresie listă).	2		
Date alternative, potrivire forme, excepţii	2		
Arbori: arbori binari (conversii liste-arbori, arbori binari de căutare, verificare proprietate arbori echilibraţi AVL, printare).	2		
Arbori: arbori Huffmann, implementare operaţii pe mulţimi. Rationator propozitional.	2		
Funcţii de ordin superior: funcţii anonime, aplicare parţială, relația dintre funcţii si date, combinatori	2		
Funcţionale pentru liste. Compilare, linkeditare, rulare	2		
Exemplificare funcţionale pentru liste (map, filter, foldr, foldl)	2		
Date infinite: evaluare leneşă, obiecte infinite, structuri circulare.	2		
Calcul Lambda: notaţie lambda, conversii, combinatori.	2		
Raţionare asupra corectitudinii programelor: inducţie structurală, echivalenţa funcţiilor, inducţie pe numărul de noduri.	2		
Elemente de programare paralela în Haskell. Exemple de aplicații și cazuri de utilizare	2		

Bibliografie *(bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziţia studenţilor într-un număr de exemplare corespunzător)*

1. G. Hutton. **Programming in Haskell, 2nd edition** Cambridge University Press, 2016
2. M. Lipovaca. **Learn You a Haskell for Great Good.** No Starch Press, 2011.
3. I.A. Letia, L.A. Negrescu, L. Negrescu. **Programare functionala, vol. I.** Ed. Albastra, Cluj-Napoca, 2006

8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observaţii
Obiecte Lisp, evaluarea formelor, funcţii Lisp primitive.	2	(Fizic/online) Rezolvare de exerciții și probleme, implementare de funcții pe calculator, trasare de algoritmi	
Reprezentare internă, controlul evaluării, definirea funcţiilor. Recursivitate şi iteraţie.	2		
Expresii LAMBDA, funcţii de ordin superior, mapare.	2		
Liste de asociaţie, proprietăţi	2		
Potrivirea şabloanelor. Prelucrări simbolice.	2		
Recapitulare programare în Lisp pentru colocviul de laborator.	2		

Colocviu de laborator (Programare în Lisp).	2		
Definirea funcţiilor Haskell, ML. Recursivitate.	2		
Operaţii pe liste.	2		
Operaţii pe arbori, grafuri.	2		
Funcţii de ordin superior Haskell, ML	2		
Evaluare leneşă.	2		
Calcul lambda	2		
Colocviu de laborator (Programare ML, Haskell).	2		

Bibliografie *(bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziţia studenţilor într-un număr de exemplare corespunzător)*

1. I.A. Leţia, E.Şt. Chifu, C. Cenan. **Programare funcţională**. Îndrumător de laborator, Casa cărţii de ştiinţă, 1999.
2. A. Cumming A gentle introduction to ML (tutorial online)

** Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatorilor reprezentativi din domeniul aferent programului

In concordanta cu obiectivele de business ale companiilor de a dezvolta produse software robuste, scalabile, cu numar minim de erori si tim de dezvoltare cat mai redus, cursul evidentiaza tehnicile specifice programarii functionale care au patruns si patrund in limbajele (multiparadigma) moderne, precum si pe corectitudinea dezvoltarii programelor. Sunt prezentate metode formale pentru verificarea corectitudinii programelor. Continutul disciplinei este in concordanta cu continutul unor cursuri similare ale altor universitati (limbaje de programare, Haskell paralel si concurent).

10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Intelegerea conceptelor specifice paradigmei de programare functionala și bazei sale teoretice. Capacitatea de a demonstra formal corectitudinea programelor	Examen final: - Fizic: examen scris - Online: Test Moodle	50%
Laborator	Abilitatea de a dezvolta cod Abilititatea de a identifica si corecta erori de programare.	Teste de laborator	50%

Standard minim de performanţă:
Abilitatea de a dezvolta cod in limbaje functionale.
Conditii de promovare: Nota ≥ 5

Data completării:	**Titulari**	**Titlu Prenume NUME**	**Semnătura**
	Curs	Conf.dr.ing. Radu Razvan Slavescu	
	Aplicații	Prof.dr.ing. Camelia Pintea	
		Ing. Vanessa Mercea	
		Ing. Sara Popa	
		Ing. Loredana Coroama	

Data avizării în Consiliul Departamentului	Director Departament Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan Prof.dr.ing. Liviu Miclea