

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea „Politehnica” din Timișoara
1.2 Facultatea ² / Departamentul ³	Automatică și Calculatoare / Calculatoare
1.3 Catedra	-
1.4 Domeniul de studii	Calculatoare și Tehnologia informației
1.5 Ciclu de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare / inginer

2. Date despre disciplină

2.1 Denumirea disciplinei	Concepte fundamentale ale limbajelor de programare						
2.2 Titularul activităților de curs	Prof. dr. ing. Horia Ciocârlie						
2.3 Titularul activităților de seminar	Asist. ing. Mirella Mioc, Asist. dr. ing. Oana Căuș						
2.4 Anul de studiu	2	2.5 Semestrul	2	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei	Opțională

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 Total ore din planul de învățământ	94	din care:3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					15
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					8
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					15
Tutoriat					7
Examinări					3
Alte activități					
3.7 Total ore studiu individual	38				
3.8 Total ore pe semestru	104				
3.9 Numărul de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Programarea Calculatoarelor, Tehnici de Programare, Programarea orientată pe obiecte
4.2 de competențe	<ul style="list-style-type: none"> Cunoștințe de matematică elementară (la nivel de liceu) Cunoștințe de programare algoritmică Cunoștințe de programare orientată pe obiecte

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none"> Sală mare, Materiale suport: laptop, proiector, tablă.
5.2 de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> Laborator cu 15-20 calculatoare – Mediu de programare pentru limbajele LISP și ML, tablă

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 (Anexa3);

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina;

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului;

6. Competențe specifice acumulate

Competențe profesionale ⁴	<ul style="list-style-type: none"> • Operarea cu fundamente științifice, ingineresti și ale informaticii • Proiectarea componentelor hardware, software și de comunicații • Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor • Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații • Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații
Competențe transversale	<ul style="list-style-type: none"> • Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională

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

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Studiul conceptelor fundamentale încorporate în limbajele de programare (LP) și dezvoltarea acestor concepte odată cu evoluția LP; • Dobândirea unei imagini de ansamblu asupra domeniului limbajelor de programare
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Sistematizarea LP; • Aprecierea calităților și lipsurilor unui LP; • Învățarea ușoară și sistematică a unui nou LP; • Utilizarea eficientă a oricărui LP; • Selectarea corectă a LP potrivite pentru o anumită aplicație; • Proiectarea unui nou LP sau a unui subset (extensie) a unuia existent. • Deprinderi practice pentru programarea funcțională

8. Conținuturi

8.1 Curs	Număr de ore	Metode de predare
1. Introducere. Calitățile unui limbaj de programare. Clasificări 1.1 Limbajul de programare în cadrul procesului de dezvoltare software 1.2 Criterii pentru evaluarea unui limbaj de programare 1.3 Cele trei familii de limbaje de programare: imperative, funcționale, declarative 1.4 Programarea secvențială și programarea concurrentă 1.5 Scurt istoric al dezvoltării limbajelor de programare	2	Prelegere, conversație, explicație, exemplificare.
2. Reprezentarea formală a limbajelor de programare 2.1 Sintaxa 2.2 Semantica	2	
3. Implementarea limbajelor de programare 3.1 Interpretarea 3.2 Translatarea 3.3 Considerente comparative 3.4 Depalii privind procesul de compilare	2	
4. Atributele entităților unui limbaj de programare 4.1 Domeniul variabilelor 4.2 Durata de existență a variabilei. Alocarea memoriei 4.3 Valoarea variabilei. Tipul variabilei	2	
5. Transmiterea datelor ca parametri 5.1 Transmiterea prin adresă, prin copiere, prin nume 5.2 Transmiterea datelor ca parametri, în diverse limbaje de programare 5.3 Transmiterea subprogramelor ca parametri 5.4 Subprograme generice	2	
6. Tipuri de date – prezentare generală 6.1 Tipuri predefinite. Tipuri definite de programator. 6.2 Tipuri scalare. Tipuri de date structurate 6.3 Tipul pointer 6.4 Compatibilitatea tipurilor 6.5 Sistemul de tipuri al limbajului Pascal, al limbajului C, al limbajului Ada, al limbajului Lisp 6.6 Comparație. Limbaje puternic tipizate	2	
7. Tipuri de date abstracte 7.1 Descrierea datelor abstracte 7.2 Date abstracte în câteva limbaje de programare moderne: trecere în revistă 7.3 Date abstracte în Modula 2, Ada, C++	4	

⁴ Aspectul competențelor profesionale va fi tratat cf. Metodologiei OMECTS 5703/18.12.2011. Se vor prelua competențele care sunt precizate în Registrul Național al Calificărilor din Învățământul Superior RNCIS (http://www.rncis.ro/portal/page?_pageid=117,70218&_dad=portal&_schema=PORTAL) pentru domeniul de studiu de la pct. 1.4, programul de studii de la pct. 1.6 din această fișă și materia în cauză

8. Limbaje orientate pe obiecte 8.1 Programarea orientată pe obiecte 8.2 Programarea orientată pe obiecte în limbajele Java, C#, Lisp	4	
9. Structuri de control în limbajele de programare 9.1 Structuri de control la nivel de instrucțiune 9.2 Subprograme 9.3 tratarea excepțiilor	4	
10. Bazele teoretice ale programării funcționale 10.1 Lambda calculus 10.2 Evaluarea leneșă 10.3 Funcții de ordin superior 10.4 Tipuri; polimorfism	4	

Bibliografie

1. Carlo Ghezzi, Mehdi Jarayeri, *Programming Languages*, John Wiley 1987.
2. Ellis Horowitz, *Fundamentals of Programming Languages*, Computer Science Press, 1984.
3. Horia Ciocârlie, *Universul limbajelor de programare*, Editura Eurostampa, 2011.

8.2 Seminar/laborator	Număr de ore	Metode de predare
1. Despre LISP. Introducere în LISP	2	Expunere temă, discuții, întrebări, rezolvare pe calculator, a 1-2 probleme.
2. Crearea de noi proceduri și realizarea evaluărilor condiționate	2	
3. Recursivitate	2	
4. Primitivele MAPCAR și APPLY. Cicluri	2	
5. Definirea procedurilor anonime cu LAMBDA	2	
6. Liste de asociații	2	
7. PRINT și READ	2	
8. Parametrii opționali, macrouri, funcții cu manevrare directă asupra listelor	2	
9. Lucrul cu matrici	2	
10. Lucrul cu celule de memorie	2	
11. Introducere în ML. Tipuri de bază. Variabile. Funcții	2	
12. Potrivirea de șabloane. Tipuri complexe	2	
13. Tipuri recursive	2	
14. Recuperări	2	

Bibliografie

1. P.H. Winston, B.K.P. Hein, *LISP*, Second Edition, Addison-Wesley, 1984.
2. Ileana Streinu, *LISP*, Editura Științifică și Enciclopedică, 1986.
3. Chris Reade, *Elements of Functional Programming*, Addison-Wesley, 1989.

9. Corelarea conținutului disciplinei cu cerințele specialiștilor din domeniu și cu așteptările angajatorilor reprezentativi

Cunoștințele predate sunt importante pentru dobândirea unei imagini de ansamblu asupra domeniului limbajelor de programare. Se ating următoarele scopuri:
<ul style="list-style-type: none"> • Sistematizarea LP; • Aprecierea calităților și lipsurilor unui LP; • Învățarea ușoară și sistematică a unui nou LP; • Utilizarea eficientă a oricărui LP.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Subiecte teoretice și exerciții care vizează înțelegerea conceptelor, problematicii și aplicării rezultatelor teoretice	Examen scris	40 %
	Subiecte aplicative constând în rezolvarea unor probleme în limbajele de programare funcționale LISP sau ML	Examen scris	25 %
10.5 Seminar /laborator	Rezolvarea problemelor corespunzătoare lucrărilor de laborator	Prezentarea rezolvărilor, răspunsuri la întrebări	20 %
	Teme de casă	Prezentarea rezolvărilor, răspunsuri la întrebări	10 %
	Prezența	Evidența prezenței	5 %

10.6 Standard minim de performanță (volumul de cunoștințe minim necesar pentru promovarea disciplinei și modul în care se verifică stăpânirea lui)

- Stăpânirea conceptelor fundamentale ale LP: variabilă, constantă, parametru, argument, tip de dată, programare orientată pe obiecte, programare funcțională.
- Proiectarea, testarea și executarea unui program de complexitate medie în LISP sau ML

11. Compatibilitate internațională

- Illinois Institute of Technology
- University of Southern California
- Princeton University

Data
completării

Semnătura titularului de curs

Semnătura titularilor de seminar

Prof. dr. ing. Horia CIOCĂRLIE

Asist. ing. Mirella MIOC, Asist. dr. ing. Oana CĂUȘ

.....

.....

Data avizării în departament

Semnătura directorului de departament

Prof. dr. Ing. Vladimir Ioan CREȚU

.....