

**FIȘA DISCIPLINEI****1. Date despre program**

1.1 Instituția de învățământ superior	Universitatea “Alexandru Ioan Cuza” din Iași
1.2 Facultatea	Facultatea de INFORMATICĂ
1.3 Departamentul	INFORMATICĂ
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	

2. Date despre disciplină

2.1 Denumirea disciplinei	Principii ale Limbajelor de Programare						
2.2 Titularul activităților de curs	Arusoae Andrei						
2.3 Titularul activităților de seminar	Arusoae Andrei						
2.4 An de studiu	2	2.5 Semestru	1	2.6 Tip de evaluare	EVP	2.7 Regimul disciplinei*	OP

* OB – Obligatoriu / OP – Opțional

3. Timpul total estimat (ore pe semestru și activități 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	56	din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiu după manual, suport de curs, bibliografie și altele					56
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					60
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					0
Tutoriat					0
Examinări					4
Alte activități					0
3.7 Total ore studiu individual					64
3.8 Total ore pe semestru					120
3.9 Număr de credite					4

4. Precondiții (dacă este cazul)

4.1 De curriculum	Logica pentru Informatică, Programare orientată-obiect
4.2 De competențe	Programare imperativă

5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	Sală de curs
5.2 De desfășurare a seminarului/ laboratorului	Sală de laborator



6. Competențe specifice acumulate

Competențe profesionale	C1. Definirea sintaxei limbajelor de programare C2. Definirea semanticii limbajelor de programare C3. Utilizarea unui cadru (framework) pentru definiții de limbaje de programare
Competențe transversale	CT1. Capacitatea de a proiecta propriul limbaj de programare CT2. Capacitatea de a crea un interpretor pentru un limbaj de programare CT3. Capacitatea de a învăța foarte ușor un nou limbaj de programare

7. Obiectivele disciplinei (din grila competențelor specifice acumulate)

7.1 Obiectivul general	Prezentarea într-o manieră simplă și corectă a conceptelor care intervin în definiția unui limbaj de programare și a unui cadru formal care să permită definirea de limbaje și testarea acestora.
7.2 Obiectivele specifice	La finalizarea cu succes a acestei discipline, studenții vor fi capabili să: <ul style="list-style-type: none">• Explice concepte specifice diferitelor paradigme de programare• Descrie sintaxa și semantica unui limbaj de programare• Utilizeze un cadru pentru definit limbaje de programare• Proiecteze un limbaj de programare

8. Conținut

8.1	Curs	Metode de predare	Observații (ore și referințe bibliografice)
1.	Despre limbaje de programare. Introducere. Paradigme de programare.	Curs videoproiector, tablă	2 ore, Capitolele 1, 13 din [1].
2.	Definirea limbajelor de programare: sintaxa	Curs videoproiector, tablă	2 ore, Capitolul 2 din [1].
3.	Definirea limbajelor de programare: semantica	Curs videoproiector, tablă	2 ore, Capitolul 6 din [1].
4.	Efecte secundare. I/O. Variabile locale.	Curs videoproiector, tablă	2 ore, [3].
5.	Nedeterminism. Fire de execuție.	Curs videoproiector, tablă	2 ore, Capitolul 8 din [1].
6.	Funcții. Call-by-value. Call-by-reference. Modelarea memoriei.	Curs videoproiector, tablă	2 ore, Capitolul 4 din [1]
7.	Tipuri în limbaje. Studii de caz: SIMPLE	Curs videoproiector, tablă	2 ore, [5]
8.	Instrumente pentru definit limbaje de programare	Curs videoproiector, tablă	2 ore, se discută unul dintre framework-urile: Coq,



			Maude, Isabelle, Racket, Rascal, Spoofox
9.	Semantica operațională: Small Step	Curs videoproiector, tablă	2 ore, [2]
10.	Semantica operațională: Big Step	Curs videoproiector, tablă	2 ore, [2]
11.	Paradigme: programare orientat-obiect	Curs videoproiector, tablă	2 ore, Capitolul 10 din [1]
12.	Paradigme: programare funcțională	Curs videoproiector, tablă	2 ore, Capitolul 11 din [1]
13.	Paradigme: programare logică	Curs videoproiector, tablă	2 ore, Capitolul 12 din [1]
14.	Alte tipuri de semantici: semantica axiomatică	Curs videoproiector, tablă	2 ore

Bibliografie**Referințe principale:**

[1] Maurizio Gabbrielli and Simone Martini Programming Languages: Principles and Paradigms. Springer-Verlag London Limited 2010

[2]. Programming Languages and Operational Semantics, Fernández, Maribel, 2014

[3]. K: A Semantic Framework for Programming Languages and Formal Analysis Tools, Grigore Roșu, Marktobendorf 16, NATO Science for Peace and Security

[4]. K framework: kframework.org

[5]. K Overview and SIMPLE Case Study, Grigore Rosu si Traian Florin Serbanuta, K'11, ENTCS.

[6]. Pagina cursului: <https://profs.info.uaic.ro/~arusoaie.andrei/lectures/PLP/plp.html>

8.2	Seminar / Laborator	Metode de predare	Observații (ore și referințe bibliografice)
1.	Instalarea framework-ului K. Compilarea definițiilor de limbaj: kompile. Stabilirea cadrului de lucru.	Discuții libere pe baza documentației. Rezolvarea eventualelor probleme.	2 ore, kframework.org
2.	Definirea sintaxei limbajului IMP: kast.	Rezolvarea fișei de exerciții.	2 ore
3.	Definirea semanticii limbajului IMP: krun.	Rezolvarea fișei de exerciții..	2 ore
4.	Efecte secundare, I/O, Variable locale in IMP.	Rezolvarea fișei de exerciții.	2 ore
5.	Explorarea nedeterminismului, implementarea firelor de execuție în IMP.	Rezolvarea fișei de exerciții.	2 ore
6.	Implementarea funcțiilor în IMP.	Rezolvarea fișei de exerciții.	2 ore
7.	Test de laborator	Evaluare	2 ore; testul are loc în fața calculatorului, iar evaluarea se face pe loc
8.	Alte instrumente pentru definit limbaje. Experimentare.	Demo in cadrul laboratorului pe videoproiector.	2 ore



9.	Discutarea ideilor de proiecte. Alegerea proiectului. Stabilirea unui plan.	Prezentarea pe videoproiector a ideilor de proiecte. Discuții directe.	2 ore
10.	Lucrul la proiect. Fișa cerințelor. Diagrame.	Discuții directe.	2 ore
11.	Lucrul la proiect: prezentarea intermediară.	Evaluare. Discuții directe.	2 ore
12.	Lucrul la proiect: arhitectura și implementare.	Discuții directe.	2 ore
13.	Lucrul la proiect: testare	Discuții directe.	2 ore
14.	Prezentarea finală a proiectului.	Evaluare. Discuții directe.	2 ore

Bibliografie[1]. K framework: kframework.org[2]. Pagina cursului: <https://profs.info.uaic.ro/~arusoaiie.andrei/lectures/PLP/plp.html>**9. Coroborarea conținutului disciplinei cu așteptările reprezentanților comunității, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului**

Această disciplină își propune să dezvolte capacitatea de a învăța rapid un limbaj de programare și chiar de a proiecta și implementa un interpretor pentru un limbaj. Acest lucru aduce o oarecare flexibilitate în ceea ce privește adaptarea la nevoile companiilor de profil care lucrează cu diverse limbaje, eventual foarte specifice unui domeniu.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere în nota finală (%)
10.4 Curs	50 puncte	Test	50%
10.5 Seminar/ Laborator	20 + 30 puncte	Proiect	50%

10.6 Standard minim de performanță : 50 puncte

Punctele la proiect se obțin în două etape: prezentare preliminară - 20 puncte și prezentare finală 30 puncte. Punctele de la prezentarea intermediară sunt luate în considerare doar dacă a fost făcută prezentarea finală a proiectului.

Data completării,

Titular de curs,

Titular de seminar,

Data avizării în departament,

Director de departament,