

**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	Informatică (limba română)

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 / EVP – evaluare pe parcurs (fara examen in sesiune)

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					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					30
Tutoriat					0
Examinări					4
Alte activități					0
3.7 Total ore studiu individual					44
3.8 Total ore pe semestru					100
3.9 Număr de credite					4

4. Precondiții (dacă este cazul)

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

5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	Cursul se va desfășura preponderent fizic în format față-în-față. Desfășurarea online va fi considerată parțial în funcție de oportunitate, necesitate și disponibilitatea mijloacelor specifice, nedepășind un quantum de maxim 22% din totalul cursurilor.
5.2 De desfășurare a seminarului/ laboratorului	Laboratorul se va desfășura preponderent fizic în format față-în-față. Desfășurarea online va fi considerată parțial în funcție de oportunitate, necesitate și disponibilitatea mijloacelor specifice, nedepășind un quantum de maxim 22% din totalul cursurilor.



6. Competențe specifice acumulate

Competențe profesionale	C1. Definirea sintaxei limbajelor de programare C2. Definirea semanticii limbajelor de programare C3. Utilizarea unui cadru (framework) pentru definirea limbajelor de programare
Competențe transversale	CT1. Capacitatea de a proiecta propriul limbaj de programare CT2. Capacitatea de a defini semantici formale pentru limbaje de programare CT3. Capacitatea de a crea un interpretor pentru un limbaj de programare CT4. 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 și principiilor 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	Studentii care promovează această disciplină 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. Prezentare generală.	Curs videoproiector, tablă	2 ore
2.	Coq: tipuri de date algebrice, definiții inductive, funcții, demonstrații simple.	Curs videoproiector, tablă	2 ore
3.	Inducție. Principii de inducție. Demonstrații prin inducție. Polimorfism. Funcții de nivel superior. Elemente de logică în Coq.	Curs videoproiector, tablă	2 ore
4.	Sintaxa concretă vs. Sintaxa abstractă: Backus-Naur Form, ambiguități, sintaxa limbajului IMP.	Curs videoproiector, tablă	2 ore
5.	Evaluarea programelor IMP.	Curs videoproiector, tablă	2 ore
6.	Semantica operațională: Big-step SOS.	Curs videoproiector, tablă	2 ore
7.	Semantica operațională: Small-step SOS.	Curs videoproiector, tablă	2 ore
8.	Evaluare.	Examen.	2 ore
9.	Sisteme de tipuri.	Curs videoproiector, tablă	2 ore



10.	Compilare certificată.	Curs videoproiector, tablă	2 ore
11.	Paradigme: programare orientată-obiect	Curs videoproiector, tablă	2 ore
12.	Paradigme: programare funcțională(*)	Curs videoproiector, tablă	2 ore
13.	Paradigme: programare logică(*)	Curs videoproiector, tablă	2 ore
14.	Verificare de programe: introducere.(*)	Curs videoproiector, tablă	2 ore

Bibliografie

1. **Practical Foundations of Programming Languages**, Robert Harper, Cambridge University Press 2016, <https://www.cs.cmu.edu/~rwh/pfpl/2nded.pdf>
2. **Software Foundations - Vol 2**, Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hrițcu, Vilhelm Sjöberg, Andrew Tolmach, Brent Yorgey, Online book: <https://softwarefoundations.cis.upenn.edu/current/index.html>
3. **Programming Languages: Principles and Paradigms**, Maurizio Gabbrielli, Simone Martini, Online: [http://websrv.dthu.edu.vn/attachments/newsevents/content2415/Programming_Languages - Principles and Paradigms thereads1106.pdf](http://websrv.dthu.edu.vn/attachments/newsevents/content2415/Programming_Languages_-_Principles_and_Paradigms_thereads1106.pdf)
4. **The formal semantics of Programming Languages – An Introduction**, Glynn Winskell. The MIT Press, Cambridge, Massachusetts, 1993. ISBN 978-0-262-23169-5. Online: <https://mitpress.mit.edu/books/formal-semantic-programming-languages>

8.2	Seminar / Laborator	Metode de predare	Observații (ore și referințe bibliografice)
1.	Instalare Coq. Stabilirea cadrului de lucru.	Discuții libere pe baza documentației. Rezolvarea eventualelor probleme.	2 ore
2.	Tipuri de date algerice. Funcții recursive. Tactici și demonstrații.	Rezolvarea fișei de exerciții.	2 ore
3.	Demonstrații prin inducție. Polimorfism și funcții de nivel superior.	Rezolvarea fișei de exerciții..	2 ore
4.	Sintaxa abstractă în Coq. Notatii. Adăugarea de noi construcții în IMP.	Rezolvarea fișei de exerciții.	2 ore
5.	Definirea unui interpretor pentru IMP.	Rezolvarea fișei de exerciții.	2 ore
6.	Big-step SOS: funcționalități noi pentru IMP.	Rezolvarea fișei de exerciții.	2 ore
7.	Small-step SOS: funcționalități noi pentru IMP.	Rezolvarea fișei de exerciții.	2 ore
8.	Evaluare.	Examen.	2 ore
9.	Sisteme de tipuri.	Rezolvarea fișei de exerciții.	2 ore
10.	Compilare certificată.	Rezolvarea fișei de exerciții.	2 ore



11.	Limbajul REC.	Rezolvarea fișei de exerciții.	2 ore
12.	Lambda calcul netipizat.(*)	Rezolvarea fișei de exerciții.	2 ore
13.	Definirea unui limbaj bazat pe stivă. (*)	Rezolvarea fișei de exerciții.	2 ore
14.	Verificare de programe: introducere (*)	Rezolvarea fișei de exerciții.	2 ore

Bibliografie

1. **Software Foundations - Vol 2, Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hrițcu, Vilhelm Sjöberg, Andrew Tolmach, Brent Yorgey, Online book:**
<https://softwarefoundations.cis.upenn.edu/current/index.html>
2. **The formal semantics of Programming Languages – An Introduction, Glynn Winskell. The MIT Press, Cambridge, Massachusetts, 1993. ISBN 978-0-262-23169-5. Online:**
<https://mitpress.mit.edu/books/formal-semantics-programming-languages>

(*) Activitățile marcate în acest mod pot avea o desfășurare online, conform condițiilor menționate la punctul 5, utilizând metode specifice asistate de tehnologie.

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	Examene pe parcurs	50%
10.5 Seminar/ Laborator	50 puncte	Fisa de lucru la laborator	50%
10.6 Standard minim de performanță : <ul style="list-style-type: none">• Definirea sintaxei unui limbaj de programare utilizând BNF sau un instrument dedicat• Definirea semanticii unui limbaj de programare utilizând semantică operațională• Definirea unui interpretor pentru un limbaj de programare utilizând un instrument dedicat			
Sunt 50 de puncte în total pentru temele de laborator. Pentru soluții deosebite și exerciții suplimentare studenții pot obține bonusuri la nota finală.			

Data completării,

Titular de curs,
Andrei ArusoiaieTitular de seminar,
Andrei Arusoiaie

Data avizării în departament,

Director de departament,
Dorel Lucanu