

**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	Departamentul de Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Informatică/Licențial în Informatică

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare funcțională						
2.2 Titularul activităților de curs	Conf. dr. Ștefan Ciobâcă						
2.3 Titularul activităților de seminar	Lect. dr. Andrei Arusoaie						
2.4 An de studiu	2	2.5 Semestru	2	2.6 Tip de evaluare	E	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					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					-
Examinări					4
Alte activități					-

3.7 Total ore studiu individual	56
3.8 Total ore pe semestru	116
3.9 Număr de credite	4

**4. Precondiții** (dacă este cazul)

4.1 De curriculum	Introducere în programare, Programare Orientată-Obiect
4.2 De competențe	Cunoașterea paradigmelor de programare imperativă și de programare orientată-obiect

5. Condiții (dacă este cazul)

5.1 De desfășurare a cursului	Cursul se desfășoară în condiții de liniște deplină, de atenție maximă și de respect atât pentru colegi, cât și pentru profesor. Studenții vor păstra ordinea și curățenia în sala de curs. Accesarea dispozitivelor electronice de către studenți nu este permisă, cu excepția cazurilor în care lecția necesită lucrul la calculator (e.g., pentru documentare, pentru experimentarea unui software etc) și profesorul permite explicit acest lucru.
5.2 De desfășurare a seminarului/ laboratorului	Laboratorul se desfășoară în condiții de liniște deplină, de atenție maximă și de respect atât pentru colegi, cât și pentru profesor. Studenții vor păstra ordinea și curățenia în sala de laborator. Studenții vor participa activ prin rezolvarea de exerciții, oferirea de răspunsuri etc.

6. Competențe specifice acumulate

Competențe profesionale	C1. Capabilitatea de a implementa algoritmi și structuri de date într-un stil de programare pur funcțional C2. Capabilitatea de a raționa despre funcții recursive C3. Capabilitatea de a defini și utiliza tipuri de date algebrice, parametrizate și de a identifica oportunitatea folosirii unor asemenea tipuri de date C4. Capabilitatea de a defini și utiliza funcții de ordin superior și de a identifica oportunitatea folosirii acestora C5. Capabilitatea de a programa într-un limbaj funcțional
Competențe transversale	CT1. Capabilitatea de a proiecta și implementa într-un stil funcțional algoritmi/structuri de date pentru probleme care țin de alte discipline (de exemplu, procesarea textelor, limbaje de programare, etc.) CT2. Capabilitatea de a utiliza instrumente algebrice pentru determinarea rezultatului unei expresii și/sau pentru determinarea de identități algebrice (de exemplu, asociativitatea funcției de concatenare a listelor)

7. Obiectivele disciplinei (din grila competențelor specifice acumulate)

7.1 Obiectivul general	Familiarizarea cu paradigma programării funcționale.
------------------------	--



7.2 Obiectivele specifice	<p>La finalizarea cu succes a acestei discipline, studenții vor fi capabili să:</p> <ul style="list-style-type: none">▪ Explice conceptele fundamentale din programarea funcțională cum ar fi funcțiile recursive, tipurile de date algebrice, funcțiile pure etc.;▪ Implementeze structuri de date și algoritmi într-un stil funcțional;▪ Utilizeze un limbaj funcțional pentru a rezolva probleme computaționale într-un stil funcțional.
----------------------------------	---

8. Conținut

8.1	Curs	Metode de predare	Observații (ore și referințe bibliografice)
1.	Introducere în programarea funcțională. Limbajul Haskell.	Expunere	2
2.	Tipuri de date. Funcții de ordin superior.	Expunere	2
3.	Clase de tipuri. Polimorfism. Module.	Expunere	2
4.	Monade. IO.	Expunere	2
5.	Lambda calcul netipizat.	Expunere	2
6.	Lambda calcul simplu tipizat.	Expunere	2
7.	Lambda calcul cu tipuri polimorfice.	Expunere	2
8.	Evaluare	Test Scris	2
9.	Strategii de evaluare.	Expunere	2
10.	Demonstrații ecuaționale.	Expunere	2
11.	Tipuri dependente 1.	Expunere	2
12.	Tipuri dependente 2.	Expunere	2
13.	Tipuri dependente 3.	Expunere	2
14.	Perspective. Test Scris.	Test Scris	2

**Bibliografie****Referințe principale:**

Graham Huton, Programming in Haskell, <http://www.cs.nott.ac.uk/~gmh/>.

Richard Bird: Introduction to Functional Programming using Haskell, Prentice Hall, 1998.

Miran Lipovača: Learn You a Haskell for Great Good! A Beginner's Guide. April 2011, 400 pp. ISBN-13: 978-1-59327-283-8

Mihai Gontineac, Programare funcțională - O introducere utilizând limbajul Haskell, Ed. Al Myller Iasi, 2006

Referințe suplimentare:

Limbajul Haskell: <http://www.haskell.org/haskellwiki/Haskell> .

http://www.haskell.org/haskellwiki/Haskell_in_education

8.2	Seminar / Laborator	Metode de predare	Observații (ore și referințe bibliografice)
1.	Introducere in implementarea Haskell GHC (ghci, comenzi, expresii și tipuri, definitii functii, tipuri pentru functii, incarcare din fisier)	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
2.	Functii recursive cu tipuri explicite (functii peste liste, prezentare builtin peste liste), inferența de tipuri	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
3.	Tipuri de date algebrice (Natural, Bool, MaybeInt, ListaInt, Binary-TreeInt, AVL, etc.) și functii recursive peste astfel de tipuri, lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
4.	Functii de ordin superior, evaluare leneșă, lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
5.	Tipuri de date algebrice parametrizate (builtin, definite de utilizator), clase de tipuri builtin (Eq, Ord, Fractional, Integral), functii parametrizate, lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
6.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2



7.	Lucru la proiect, test scris	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
8.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
9.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
10.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
11.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
12.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
13.	Lucru la proiect	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2
14.	Lucru la proiect, test scris	Exemplificare la tablă, exemplificare folosind video-proiector, lucru individual/în echipă la calculator	2

Bibliografie

<http://www.haskell.org/ghc/>
<http://www.haskell.org/haskellwiki/GHC>
<http://www.haskell.org/haskellwiki/Tutorials>

**9. Coroborarea conținutului disciplinei cu așteptările reprezentanților comunității, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului**

Conținutul cursului este coroborat cu tendințele din industrie de adoptare a tehnicilor de programare funcțională.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere în nota finală (%)
10.4 Curs	Înțelegerea noțiunilor și conceptelor prezentate în curs	Test scris	50%
10.5 Seminar/ Laborator	Însușirea noțiunilor parcurse în timpul laboratorului	Teste scrise	50%
10.6 Standard minim de performanță			
Test scris (cel puțin 50% din punctajul aferent) Activitate laborator (cel puțin 50% din punctajul aferent)			

Data completării

Titular de curs

Titular de seminar

Conf. dr. Ștefan Ciobâcă

Lect. dr. Andrei Arusoiaie

Data avizării în departament

Director de departament