FIŞA DISCIPLINEI

Course title/ Titlul cursului:	Proiectarea compilatoarelor				
Course code/ Codul cursului:	**Type of course/ Tipul cursului:**	**Level of course/ Nivelul cursului:**	**Year of study/ An de studiu:**	**Semester/ Semestru:**	**Number of credits/ Număr de credite:**
38060244	Obligatoriu	Studii de licență	3	5	6
Name of lecturer/ Numele profesorului:	**Titular curs/seminar/laborator/proiect:** Prof.univ.dr. Emil M. Popa/ Lector univ. dr. Alina E. Pitic				
Department/ Departament (ce coordonează disciplina):	DEPARTAMENTUL DE MATEMATICA SI INFORMATICA				

Extinderea disciplinei în planul de învăţământ				
Lecture/Curs	**Seminar**	**Lab/Laborator**	**Project/Proiect**	**Total ($NOAD_{sem}$)**
28		28		56

Bugetul de timp pentru studiu individual			
Type of activities / Denumirea activităţii	**Hours/Nr. Ore**	**Type of activities / Denumirea activităţii**	**Hours/Nr. Ore**
1. Studierea notiţelor de curs	22	**5. Pregătirea seminariilor/laboratoarelor**	20
2. Studiul după suport de curs		**6. Elaborarea referatelor, temelor individuale**	20
3. Studierea bibliografiei minimale	20	**7. Pregătirea pentru evaluările periodice**	30
4. Documentare suplimentară (internet, bibliotecă ş.a.)		**8. Participarea la consultaţii**	
Total ore alocate studiului individual $NOSI_{sem}$			112

Objectives of the course/ Obiectivele cursului:	a) Iniţierea studenţilor în teoria compilatoarelor b) Noţiuni de construcţia compilatoarelor: fazele compilării, analiza lexicala, analiza sintactica, analiza semantica. c) Prezentarea sistematica a noţiunilor şi metodelor de baza utilizate la proiectarea şi implementarea limbajelor de programare. d) Familiarizarea studenţilor cu gândirea algoritmică; e) forme de specificare a limbajelor (regulare, independente de context); a) înţelegerea modurilor de specificărilor unui limbaj nou; b) structura si funcţionarea unui translator; c) deprinderilor de realizare în grup a unui produs program corect (cu limbajele C++, C#, Java), prin parcurgerea tuturor etapelor necesare şi reflectarea lor într-o documentaţie completă. f) Aprofundarea cunoştinţelor de programare ale studenţilor	
Prerequisites/ Discipline precursoare obligatorii:	Limbaje formale şi teoria automatelor, OOP	
Course contents/ Conţinutul cursului:	Cursul 1	Noţiuni introductive şi notaţii. Tipuri de limbaje de programare. Procesoare de limbaje. Structura generala a unui compilator.

	Cursul 2 - 3	Expresii regulate. Operaţii cu expresii regulate. Automatul finit echivalent cu o expresie regulată. Tehnici de construire a automatelor finite din expresii regulate. Automat finit nedeterminist echivalent cu un automat finit determinist. Minimizarea automatelor finite.
	Cursul 4	Analiza lexicală
	Cursul 5	Analiza sintactică. Algoritmi generali de analiză sintactică top-down şi bottom-up. Algoritmul de analiză Cocke-Younger-Kasami
	Cursul 6	Algoritmi de analiză cu complexitate liniară. Algoritmul de analiza sintactică LL(k)
	Cursul 7	Algoritmul de analiza sintactică LR(k)
	Cursul 8	Gramatici de precedenţă
	Cursul 9 - 10	Analiza semantică. Gramatici de atribute. Verificarea tipurilor
	Cursul 11 - 13	Generarea şi optimizarea de cod
	Cursul 13 - 14	Gestiunea tabelei de simboluri. Tipuri de tabele Detectarea erorilor. Sursele şi corectarea erorilor. Observaţii şi concluzii
Lab/Seminary contents/ Conţinutul laboratorului/seminarului:	Laborator 1	Recapitulare noţiunilor de baza pentru mecanismele care vor fi studiate. Prezentarea principalelor componente ale unui compilator. Familiarizarea cu mediul de dezvoltare.
	Laborator 2 - 3	Modelul analizorului lexical. Generare analizoarelor lexicale. Construcţia unui semiautomat finit pornind de la o gramatica de tipul 3. Transformarea expresiilor regulate în automate finite. Stabilirea echipelor pentru temele de implementare. Împărţirea proiectelor.
	Laborator 4	Transformarea unui automat finite nedeterminist în automat finit determinist. Minimizarea automatelor finite.
	Laborator 5	Analiza sintactica. Algoritmul general de analiza top-down (cu şi fără revenire).
	Laborator 6	Algoritmi generat de analiza bottom-up. Algoritmul de analiza Cocke-Younger-Kasami
	Laborator 7	Algoritmi de analiza sintactica de complexitate liniara. Gramatici si limbaje LL(k). Implementarea operatorului $\oplus_k$.
	Laborator 8 - 9	Gramatici şi limbaje LR(k). Construcţia tabelelor de analiza LR(k)
	Laborator 10	Gramatici şi limbaje de precedenta. Construcţia matricei de precedenţă
	Laborator 11	Tabele de simboluri. Analiza tipurilor
	Laborator 12	Construcţia unui translator simplu. Interpretor de cod virtual.
	Laborator 13	Construcţia unui generator simplu de cod maşină.
	Laborator 14	Predarea şi prezentarea temelor de implementare.

Teaching methods/	**Language of instruction/ Limba de predare: Română**
Metode de predare: La curs se va folosi expunerea, explicatia, exemplificarea si conversatia frontala. La laborator se va folosi explicatia, exemplificarea, invatarea prin descoperire. Pentru curs si laborator exista suport tiparit. La curs se vor folosi si slide-uri si exemplificare pe calculator.	

Assesment methods/ Sisteme de evaluare:	Activităţi aplicative - 5%	1. Teme de curs/pondere= %(nCPC) 2. Referate de disciplină= %(nCPC) 3. Lucrări practice= 50 %(CPC)
	Proiect - 45 %	CPE (CPE – condiţionează participarea la examen)
	Examen parţial - %	(nCPE – nu condiţionează participarea la examen)
	Examen de semestru - 50 %	(condiţionează evaluarea finală)
Competenţe specifice disciplinei		
1. Competenţe privind cunoaşterea şi înţelegerea	• Cunoaşterea şi utilizarea adecvată a noţiunilor specifice • Iniţierea studenţilor în teoria calculabilităţii; • Studierea şi aprofundarea diverselor modele formale pentru sisteme de calcul. • Investigarea computaţilor posibile cu diverse combinaţii de resurse de calcul. • Introducerea a mai multor formalizări aparent diferite pentru noţiunea de algoritm;	
2. Competenţe în domeniul explicării şi interpretării	• Capacitatea de a realiza si implementa gramaticile si automatele • Studierea calculelor posibile cât şi a celor imposibile • Prezentarea de rezultate legate de calcule imposibile pe orice sistem de calcul, indiferent de dimensiune sau putere de calcul. • Familiarizarea studenţilor cu gândirea algoritmică; • Capacitatea de a planifica activitatea de salvare, refacere şi recuperare • Capacitatea de a realiza si implementa transformarea din gramatici in automate • Capacitatea de a realiza si implementa transformarea din automate in gramatici	
3. Competenţe instrumental - aplicative	Capacitatea de a proiecta şi realiza aplicaţii ale gramaticilor independente de context: cu clase, cu si fara fisier extern.	
4. Competenţe atitudinale	Dezvoltarea atitudinii pozitive faţă de muncă şi responsabilitate pentru propria pregătire profesională	
Competenţele generale sunt menţionate în Fişa specializării		
Recommended reading/ Referinţe bibliografice recomandate (max. 10):	1. Vasile Crăciunean, *Proiectarea Translatoarelor*, Sibiu, Editura Alma Mater, 2002. 2. Ralf Fabian, *Limbaje formale Teorie. Exemple. Probleme*, Editura Universităţii „Lucian Blaga", Sibiu, 2006, pag. 123, ISBN 973-739-225-6 3. Hopcroft, J.E., Motwani, R. and Ullman, J.D. - *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley 2007. 4. Emil M. Popa, *Limbaje formale, Fundamentele limbajelor de programare*, Editura „Alma Mater", Sibiu, 2003.	
More references/ Referinţe bibliografice suplimentare:	1. Emil M. Popa, *Formal Syntax and Semantics of Programming Language*, Editura „Alma Mater", Sibiu, 2004. 2. Aho,A.V., Ullman, J.D., *The Theory of Parsing Translations and Compiling*, vol.1: Parsing, vol2: Compiling, New Jersey, Prentice-Hall, inc., 1972. 3. Teodor Rus, *Mecanisme formale pentru specificarea limbajelor*, Ed. Academie Române, Bucureşti, 1983. 4. Luca Dan Şerbănaţi, *Limbaje de programare şi compilatoare*, Ed. Academiei Române, Bucureşti, 1987. 5. Ernest G. Manes, Michael A. Arbib. *Algebraic Approaches to program semantics* – Springer – Verlag New York Berlin Heidelberg London Paris Tokyo – 1986 6. Salomaa A., *Formal Languages*, New York, Academic Press, 1973. 7. Creanga I., Reischer C., Simovici D. *Introducerea algebrică in informatică*, vol. 1 şi vol. 2, Editura Tehnică 1974 8. Paun Gh., *Probleme actuale in teoria limbajelor formale*, Editura Academiei, 1983. 9. Păun Gh., *Gramatici contextuale*, Bucureşti, 1982 10. Păun Gh., *Mecanisme generative ale proceselor economice*, Ed Tehnică, Bucureşti, 1988	
Regulamentul disciplinei		

Data elaborării:
28.09.2013

Titulari disciplină:

Prof.univ.dr. Emil M. Popa/
Lector univ. dr. Alina E. Pitic

Adresa: Str. Ion Raţiu, nr. 5-7
Sibiu, 550024, România
e-mail: stiinte@ulbsibiu.ro
web: stiinte.ulbsibiu.ro

Tel: +40 (269) 216 642
Fax: +40 (269) 216 617