SYLLABUS

1. Information regarding the programme

1.1 Higher education institution	**Babeş Bolyai University**
1.2 Faculty	**Faculty of Mathematics and Computer Science**
1.3 Department	**Department of Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Bachelor**
1.6 Study programme / Qualification	**Computer Science**

2. Information regarding the discipline

2.1 Name of the discipline	**Functional and Logic Programming**						
2.2 Course coordinator	**Prof.Dr. Horia F. Pop**						
2.3 Seminar coordinator	**Assist.Dr. Radu Găceanu**						
2.4. Year of study	**2**	2.5 Semester	**3**	2.6. Type of evaluation	**C**	2.7 Type of discipline	**Compulsory**

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1 lab
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14
Time allotment:					hours
Learning using manual, course support, bibliography, course notes					23
Additional documentation (in libraries, on electronic platforms, field documentation)					13
Preparation for seminars/labs, homework, papers, portfolios and essays					23
Tutorship					4
Evaluations					20
Other activities:					-

3.7 Total individual study hours	83
3.8 Total hours per semester	125
3.9 Number of ECTS credits	5

4. Prerequisites (if necessary)

4.1. curriculum	• Fundamentals of Programming • Mathematical Foundations of Computer Science
4.2. competencies	• Average programming skills in a high level programming language

5. Conditions (if necessary)

5.1. for the course	• Students will attend the course with their mobile phones shut down
5.2. for the seminar /lab	• Students will attend the lab with their mobile phones shut down

activities	• Laboratory with computers; high level declarative programming language environment (Common Lisp, Turbo Prolog)

6. Specific competencies acquired

Professional competencies	• Define notions, concepts, theories and models in functional and logic programming and their appropriate use in programming activities • Use knowledge of programming paradigms to model and solve various real-world problems • Identify and apply the declarative concepts, methods and theories to solve typical programming problems • Ability to use functional and logic programming environments in regular problem solving
Transversal competencies	• Execution of the tasks under specified requirements and the deadlines imposed, according to professional ethics and moral conduct • Manage tasks according to the generally established objectives • Permanent informing and documentation in the topic of the course • Concern for improving the results of professional activity by personal involvement in the activities

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Get accustomed with basic notions, concepts, theories and models of new programming paradigms (functional and logic programming)
7.2 Specific objective of the discipline	• Get accustomed with a programming language for each of these paradigms (Common Lisp and Turbo Prolog) • Acquire the idea of using these programming paradigms based on the applications' necessities • Assure the necessary base for approaching certain advanced courses • Ability to apply declarative programming techniques to different real life problems • Ability to model phenomena using declarative techniques • Improved programming abilities using the declarative paradigm

8. Content

8.1 Course	Teaching methods	Remarks
1. Basic elements of Prolog. Facts and rules in Prolog. Goals. The control strategy in Prolog. Variables and composed propositions. Anonymous variables. Rules for matching. The flux model. Sections of a Prolog program. Examples	Exposure: description, explanation, examples, discussion of case studies	
2. The Prolog program. Predefined domains. Internal and external goals. Multiple arity predicates. The IF symbol (Prolog) and the IF instruction (other languages). Compiler directives. Arithmetic expressions and comparisons. Input/output operations. Strings	Exposure: description, explanation, examples, discussion of case studies	
3. Backtracking. The backtracking control. The "fail" and "!"(cut) predicates. Using the "!" predicate. Type of cuts. The "not" predicate. Lists in Prolog. Recursion. Examples for backtracking in Prolog. Finding all solutions in	Exposure: description, explanation, examples, discussion of case studies	

the same time. Examples of predicates in Prolog. Non-deterministic predicates		
4. Composed objects and functors. Unifying composed objects. Arguments of multiple types; heterogeneous lists. Comparisons for composed objects. Backtracking with cycles. Examples of recursive procedures. The stack frame. Optimization using the "tail recursion". Using the "cut" predicate in order to keep the "tail recursion".	Exposure: description, explanation, examples, discussion of case studies	
5. Recursive data structures. Trees as data structures. Creating and traversing a tree. Search trees. The internal database of Prolog. The "database" section. Declaration of the internal database. Predicates concerning operations with the internal database.	Exposure: description, explanation, examples, discussion of case studies	
6. Advanced issues of Backtracking in Prolog. Files management in Prolog.	Exposure: description, explanation, examples, proofs, debate, dialogue	
7. Graded paper in Logic Programming	Written test	
8. Programming and programming languages. Imperative programming vs. declarative programming. Introduction. The importance of the functional programming as a new programming methodology. History and presentation of LISP	Exposure: description, explanation, examples, discussion of case studies	
9. Basic elements in Lisp. Dynamic data structures. Syntactic and semantic rules. Functions' classification in Lisp. Primitive functions in Lisp. Basic predicates in Lisp.	Exposure: description, explanation, examples, discussion of case studies	
10. Predicates for lists; for numbers. Logic and arithmetic functions. Defining user functions. The conditional form. The collecting variable method. Examples	Exposure: description, explanation, examples, discussion of case studies	
11. Symbols' managing. Other functions for lists' accessing. OBLIST and ALIST. Destructive functions. Comparisons. Other interesting functions. Examples	Exposure: description, explanation, examples, discussion of case studies	
12. Definitional mechanisms. The EVAL form. Functional forms; the functions FUNCALL and APPLY. LAMBDA expressions, LABEL expressions. Generators, functional arguments. MAP functions. Iterative forms. Examples	Exposure: description, explanation, examples, discussion of case studies	
13. Other elements in Lisp. Data structures. Macro-definitions. Optional arguments. Examples	Exposure: description, explanation, examples, discussion of case studies	
14. Graded paper in Functional Programming	Written test	

Bibliography

CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012

POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, Cluj-Napoca, 2003

* * *, Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp.

* * *, Documentatia produselor: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog.

http://www.ifcomputer.com/PrologCourse, Lecture on Prolog

http://www.lpa.co.uk, Logic Programming		
8.2 Laboratory	Teaching methods	Remarks
Lab 1: Recursive algorithms in Pseudocode	Explanation, dialogue, testing data discussion, case studies	Deliver problem at lab 1 Receive problem for lab 2
Lab 2: Lists in Prolog	Explanation, dialogue, testing data discussion, case studies	Deliver problem at lab 2 Receive problem for lab 3
Lab 3: Trees in Prolog. Lists management in Prolog.	Explanation, dialogue, testing data discussion, case studies	Deliver problem at lab 3 Receive problem for lab 4
Lab 4: Backtracking in Prolog	Explanation, dialogue, testing data discussion, case studies	One hour Deliver problem at lab 4 Receive problem for lab 5
Lab 4: Practical test in Prolog	Practical test	One hour
Lab 5: Recursive programming in Lisp	Explanation, dialogue, testing data discussion, case studies	Deliver problem at lab 5 Receive problem for lab 6
Lab 6: Using MAP functions in Lisp.	Explanation, dialogue, testing data discussion, case studies	Deliver problem at lab 6 Receive problem for lab 7
Lab 7: Iterative programming in Lisp	Explanation, dialogue, testing data discussion, case studies	One hour Deliver problem at lab 7
Lab 7: Practical test in Lisp	Practical test	One hour
Bibliography CZIBULA G., POP H.F., Elemente avansate de programare in Lisp si Prolog. Aplicatii in Inteligenta Artificiala, Editura Albastra, Cluj-Napoca, 2012 POP H.F., SERBAN G., Programare in Inteligenta Artificiala - Lisp si Prolog, Editura Albastra, Cluj-Napoca, 2003 * * *, Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp. * * *, Documentatia produselor: Turbo Prolog 2.0, Logic Explorer, Sicstus Prolog. http://www.ifcomputer.com/PrologCourse, Lecture on Prolog http://www.lpa.co.uk, Logic Programming		

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- The course respects the IEEE and ACM Curriculla Recommendations for Computer Science studies;
- The course exists in the studying program of all major universities in Romania and abroad;
- The content of the course is concordant with partial competencies for possible occupations from the Grid 1 - RNCIS.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	- know the basic principle of the domain; - apply the course concepts - problem solving	Written test in Logic Programming	30%
		Written test in Functional Programming	30%

10.5 Seminar/lab activities	- be able to implement course concepts and algorithms - apply techniques for different classes of programming languages	Programs documentation and delivery	20%
		Practical test in Prolog	10%
		Practical test in Lisp	10%
10.6 Minimum performance standards			
➢ Each student has to prove that (s)he acquired an acceptable level of knowledge and understanding of the subject, that (s)he is capable of stating these knowledge in a coherent form, that (s)he has the ability to establish certain connections and to use the knowledge in solving different problems. ➢ For final passing of the course, the following minimal criteria apply collectively: at least grade 4 (from a scale of 1 to 10) at both written tests; successful delivery of 5 out of 10 lab problems; at least grade 5 (from a scale of 1 to 10) computed as final grade average.			

Date	Signature of course coordinator	Signature of seminar coordinator
30.04.2013	Prof.Dr. Horia F. POP	Assist.Dr. Radu GĂCEANU
Date of approval		Signature of the head of department
.....		Prof.Dr. Bazil Pârv