

UP Faculty of Sciences		Subject Description	Page Nr.: 1/2
2. Course Title: Non-imperative languages			
3. Code:		4. Type (lecture, lab etc.): lecture+lab	
5. Nr. of hours weekly: 3 (2lect+1lab)		5	
7. Preconditions (max. 3 preceeding course): —			
8. How often is it offered: <input type="checkbox"/> Fall semester, <input checked="" type="checkbox"/> Spring semester, <input type="checkbox"/> both semesters			
9. Maximal nr. of students: 38 per groups			
10. Lecturer responsible (faculty, institution and dept.): Dr. Kilián Imre (Fac. of Sciences, Institute of Mathematics and Informatics, Department of Information Technology and Biorobotics)			
11. Lecturers with their procentual rates:		Dr. Kilián Imre	100%
12. Course language: English			
13. Learning outcomes: The purpose of the subject is that students become familiar with the paradigm of non-imperative programming, along with its languages, its approach and programming methodology, based primarily on the Prolog programmin language family. Students, completing the course: <i>know</i> the elements of the most important non-imperative programming languages (Prolog and LISP), and their application areas. <i>able</i> to express and implement simple algorithms in Prolog <i>able</i> to give Prolog constrained logic (CLP) solutions for simple problems.			
14. A tantárgyi program, 13 oktatási hétre osztott részletes tematika: 1. hét: Overview of non imperative programming languages, and classification by paradigm and application area. 2. hét: Overview of LISP, as a programming language. 3. hét: Mathematical foundations of Prolog. Resolutional theorem proving for Horn-clauses. 4. hét: The procedural semantics of Prolog. Variables and constants. Compound data structures. Pattern matching and backtracking. 5. hét: Pruning the search tree, Cut. Implementing imperative(-like) constructs with PROlog. Recoursive and backtracking loops. 6. hét: List management algorithms 7. hét: Open data structures. Open (difference) lists. 8. hét: Difference list handling algorithms. 9. hét: The most important built-in procedure packages. (arithmetics, I/O, program-state modification) 10. hét: The SWI-Prolog programming environment 11. hét: Metaprogramming and its applications. The 4 gate calling mode. Debugging. 12. hét: Connection of SWI Prolog to other programing languages. (C++, Java). SWISH a Prolog server. 13. hét: Constraint logic programming.			
15. Special tasks during the semester:			
16. Description of evaluation: <ul style="list-style-type: none"> oral exam + tests, an average is calculated, in case of doubt test results are prioritized 			
17. Required reading: [1] Clockshin-Mellish: Programming in Prolog Springer-Verlag, Fifth Edition. 2003.			

UP Faculty of Sciences		Subject Description		Page Nr.: 2/2
[2] Peter Seibel: Practical Common LISP Apress, Singer Nature, 2005.				
18. Proposed reading:				
The description has been made:	2017. március 15	Made by:		
			Dr. Kilián Imre lecturer	
			Approved by:	