

STUDY MODULE DESCRIPTION FORM		
Name of the module/subject Multiparadigm programming		Code 1010334591010337136
Field of study Information Engineering	Profile of study (general academic, practical) (brak)	Year /Semester 5 / 9
Elective path/specialty Information Technologies	Subject offered in: Polish	Course (compulsory, elective) obligatory
Cycle of study: First-cycle studies	Form of study (full-time, part-time) part-time	
No. of hours Lecture: 8 Classes: - Laboratory: 8 Project/seminars: -		No. of credits 3
Status of the course in the study program (Basic, major, other) (brak)		(university-wide, from another field) (brak)
Education areas and fields of science and art technical sciences Technical sciences		ECTS distribution (number and %) 3 100% 3 100%
Responsible for subject / lecturer: dr inż. Krzysztof Zwierzyński email: Krzysztof.Zwierzynski@put.poznan.pl tel. +48 61 665 3755 Faculty of Electrical Engineering ul. Piotrowo 3A 60-965 Poznań		
Prerequisites in terms of knowledge, skills and social competencies:		
1	Knowledge	Student has basic knowledge of logic, theory of recursive functions, imperative and declarative programming, object-oriented programming, data bases, operating systems and computer networks.
2	Skills	Student is able to acquire information from literature, data bases and other sources; student is able to integrate acquired information, to interpret it, to draw conclusions and to formulate and justify judgments. Student is able to communicate in English and to read descriptions and manuals of software tools, applications and similar documents.
3	Social competencies	Student understands the necessity and possibility of continuous education and development of different skills (linguistic, professional, personal and social). Student understands a responsibility associated to his own work. Student is able to adhere to team work rules and to take responsibility for cooperative tasks.
Assumptions and objectives of the course: an overview of computation paradigms and presentation of basic concepts, techniques and programming abstractions. Acquiring the skills of selecting an appropriate computation model for a given problem; gaining the practice in multiparadigm programming.		
Study outcomes and reference to the educational results for a field of study		
Knowledge: 1. Student has organized knowledge with theoretical foundations of basic program constructions, algorithm implementations, paradigms and programming styles, software verification methods, formal languages, compilers, platforms. - [[K_W05]]		
Skills: 1. Student is able to use software platforms and environments for simple programs encoding, running and testing in imperative, object-oriented and declarative programming languages. - [[K_U10]]		
Social competencies: 1. Student understands the importance of stringent accomplishment of a given project with proper notation standards, proper language. Student understands the importance of keeping deadlines. - [[K_K07]]		
Assessment methods of study outcomes		
Lecture. Written test based on lecture (basic concepts and simple tasks). Laboratory. Students' marks are based on continuous assessment of their programming activity and results of two written tests (creation of simple programs).		

Course description		
<p>Lectures. Declarative computation paradigm. Concepts and techniques of the functional and deterministic logic programming. Iterative and recursive programming, metaprograming, abstract data types. Declarative concurrency. Relational programming and data bases. Integrating a logic programming paradigm and a constraint programming paradigm.</p> <p>Laboratory. Creation of simple programs with multiparadigm technigues, particularly functional programming and declarative concurrency in Erlang language.</p> <p>Course update 2017: programming in Erlang language, new techniques of constraint programming - redundant constraints and reified constraints.</p> <p>Teaching methods:</p> <ul style="list-style-type: none"> - lectures supported by slides and examples presented on the table - laboratories - writing programs by individual students, discussion of proposed solutions, a usage of tools enabling students to perform taksks at home. 		
<p>Basic bibliography:</p> <ol style="list-style-type: none"> 1. Armstrong J.: Programming Erlang. The Pragmatic Programmers, 2013 2. Haber F.:LEARN YOU SOME ERLANG FOR GREAT GOOD! A BEGINNER&#39;S GUIDE (on-line learnyousomeerlang.com) 3. Roy P. van, Haridi S.: Concepts, Techniques and Models of Computer Programming, The MIT Press, 2004 		
<p>Additional bibliography:</p> <ol style="list-style-type: none"> 1. Cesarini F., Thompson S.: Erlang Programming. O&#39;Reilly Media, 2009 2. Kowalski R.: Logic for problem solving, North-Holland, 1979 3. Zwierzyński K.T., Meissner A., Niwińska M., A Method Involving Constraint Programming for Generating Integral Graphs without +1 in the Spectrum. A Case Study, Studies in Automation and Information Technology, Vol. 35, PTPN, Poznań, 2010, s. 105-114. 4. Meissner A., Niwińska M., Zwierzyński K., Computing the Irregularity Strength of Connected Graphs by Parallel Constraint Solving in the Mozart System, Lecture Notes in Computer Science, Vol. 4967, Springer, Berlin-Heidelberg, 2008, s. 1096-1103. 		
Result of average student's workload		
Activity	Time (working hours)	
1. Lecture	8	
2. Laboratory	8	
3. Preparation for laboratory and tests	45	
Student's workload		
Source of workload	hours	ECTS
Total workload	75	3
Contact hours	30	1
Practical activities	45	2