1.3. Module/ course form

To be completed by Course Team	Module name : ***Formal languages and compilation methods***				Module code:		
	Course name: *Formal languages and compilation methods*				Course code:		
	Faculty: **Institute of Applied Informatics**						
	Field of study:						
	Mode of study :		Learning profile:		Speciality:		
	Year/ semester: 5-th		Module/ course status:		Module/ course language:		
	Type of classes	lecture	lessons	lab	project	tutorial	other (please specify)
	Course load	**15h**		**15h**			

Module/ course coordinator	Stefan Sokołowski
Lecturer	Stefan Sokołowski
Module/ course objectives	Acquaintance with formal grammars and formal acceptors, as used in computer science; and a bird's eye view on principle of operation of modern compilers.
Entry requirements	The course on *Foundations of Programming* (Podstawy programowania)

LEARNING OUTCOME

Nr	LEARNING OUTCOME DESCRIPTION	Learning outcome reference
1	Reading and putting together context-free grammars for simple formal languages	
2	Applying and setting up finite automata and stack machines	
3	Programming a recursive descent compiler	
4	Programming a simple precedence parser	
5	Target code generation -- low-level (machine code) programming	

CURRICULUM CONTENTS

Lecture

1. **Formal languages and grammars:**

the notion of grammar, defining languages by grammars

regular and context-free languages
the existence or non-existence of a grammar to a given language
syntax of programming languages

2. **Models of theoretical computing devices:**

finite automata and their relation to regular languages
stack machines and their relation to context-free langauages

3. **Lexical analysis (scanning):**

the notion of lexem
text scanners based on finite automata
computer implementation of a scanner

4. **Syntax analysis (parsing):**

construction of a parse tree
recursive top-down parsing
precedence table-driven bottom-up parsing

5. **Low-level (machine code) programming**

6. **Outline of target code generation**

context-dependent features in programming languages
memory management
compilation from a high-level language to an abstract stack machine code
compilation from an abstract stack machine code to machine code

7. **Semi-automatic compiler construction: LEX (FLEX) and YACC (BISON)**

Tutorial

in Polish:
http://student.pwsz.elblag.pl/~stefan/Dydaktyka/2012-2013/JezForm/

Basic literature	Gries D. *Compiler construction for digital computers,* John Wiley & Sons, 1971
Additional literature	

Teaching methods	Lecture and computer laboratory exercises.

Assessment method	Learning outcome number
A number of computerized class tests	
Written exam	

Form and terms of an exam	

STUDENT WORKLOAD

	Number of hours
Participation in lectures	15
Independent study of lecture topics	5
Participation in tutorials, labs, projects and seminars	15
Independent preparation for tutorials*	25
Preparation of projects/essays/etc. *	
Preparation/ independent study for exams	10
Participation during consultation hours	5
Other	
TOTAL student workload in hours	75
Number of ECTS credit per course unit	**3 ECTS**
Number of ECTS credit associated with practical classes	40 **1,6 ECTS**
Number of ECTS for classes that require direct participation of professors	35 **1,4 ECTS**