Module: Principles of Compilers

Level	Bachelor	**Short Name**	PC
Responsible Lecturers	Harder, Andreas, Dr.		
Department, Facility	Electrical Engineering and Computer Science		
Course of Studies	Information Technology, Bachelor		
Compulsory/elective	Compulsory	**ECTS Credit Points**	5
Semester of Studies	7	**Semester Hours per Week**	4
Length (semesters)	1	**Workload (hours)**	150
Frequency	WiSe	**Presence Hours**	60
Teaching Language	English	**Self-Study Hours**	90
The following section is filled only if there is **exactly one** module-concluding exam.			
Exam Type	Written Exam	**Exam Language**	English
Exam Length (minutes)		**Exam Grading System**	One-third Grades
Learning Outcomes	The students recognize the connections of the theoretical principles from automata theory and formal languages to compiler construction and to the tasks of the individual phases of compiling a programming language. The module covers the following phases of a classical compiler: scanner, parser, semantic analysis. Common methods are presented for each phase, especially LL-parsers. In the exercise, the methods presented are deepened with small examples. In some parts, a parser generator (typically ANTLR) will be used by the students to build their own compiler and to evaluate the practical use of a parser generator.		
Participation Prerequisites	Knowledge of C-Programming, operating systems, and mathematics (in particular basics of logic and set theory)		
The previous section is filled only if there is **exactly one** module-concluding exam.			
Consideration of Gender and Diversity Issues	✓ Use of gender-neutral language (THL standard) ✗ Target group specific adjustment of didactic methods ✓ Making subject diversity visible (female researchers, cultures etc.)		
Applicability			
Remarks			

Module Course: Principles of Compilers (Lecture)

(of Module: Principles of Compilers)

Course Type	Lecture	**Form of Learning**	Presence
Mandatory Attendance	no	**ECTS Credit Points**	3
Participation Limit		**Semester Hours per Week**	3
Group Size		**Workload (hours)**	90
Teaching Language	English	**Presence Hours**	45
Study Achievements ("Studienleistung", SL)		**Self-Study Hours**	45
SL Length (minutes)		**SL Grading System**	
The following section is filled only if there is a course-specific exam.			
Exam Type		**Exam Language**	
Exam Length (minutes)		**Exam Grading System**	
Learning Outcomes			
Participation Prerequisites			
The previous section is filled only if there is a course-specific exam.			
Contents	• Introduction: Compilers and Interpreters; Structure of Compilers • Lexical analysis: theory; tokens, regular expressions and finite automata. • Lexical analysis: lexical analyzer; Lex; symbol table management. • Syntax analysis: theory; context-free grammars, parse trees, pushdown automata, construction of LL(1)-parsers. • Syntax analysis: deterministic syntax analysis, FIRST and FOLLOW, LL and LR grammars. • Deterministic top-down syntax analysis: recursive descent. • Semantic analysis • Automatic code generation (ANTLR)		
Literature	Terence Parr: Language Implementation Patterns, Pragmatic BookshelfTerence Parr: The Definitive ANTLR Reference: Building Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman: Principles of Compilers, Addison Wesley		
Remarks			

Module Course: Principles of Compilers (Exercises)

(of Module: Principles of Compilers)

Course Type	Exercise	**Form of Learning**	Presence
Mandatory Attendance	no	**ECTS Credit Points**	2
Participation Limit		**Semester Hours per Week**	1
Group Size	12	**Workload (hours)**	60
Teaching Language	English	**Presence Hours**	15
Study Achievements ("Studienleistung", SL)		**Self-Study Hours**	45
SL Length (minutes)		**SL Grading System**	
The following section is filled only if there is a course-specific exam.			
Exam Type		**Exam Language**	
Exam Length (minutes)		**Exam Grading System**	
Learning Outcomes			
Participation Prerequisites			
The previous section is filled only if there is a course-specific exam.			
Contents			
Literature	See literature for the lecture		
Remarks			