

Teilmodul		TM-Kurzbezeichnung
DCMC: Compilerbau		DCMC
Verantwortliche/r	Fakultät	
Prof. Dr. Florian Heinz	Informatik und Mathematik	
Lehrende/r / Dozierende/r	Angebotsfrequenz	
Prof. Dr. Florian Heinz	unregelmäßig wiederholende Lehrveranstaltung	
Lehrform		
Seminaristischer Unterricht mit integrierten Übungen (gesamt 4 SWS)		

Studiensemester gemäß Studienplan	Lehrumfang [SWS oder UE]	Lehrsprache	Arbeitsaufwand [ECTS-Credits]
6. / 7.	4 SWS	englisch	5

Zeitaufwand:

Präsenzstudium	Eigenstudium
60h	90h

Studien- und Prüfungsleistung
Kl. u./o. StA u./o. mdl. LN

Inhalte
<p>A compiler in the narrow sense is a computer program that translates some source code into a machine program for a specific processor architecture. Building a compiler is not a trivial task and consists of several steps that include lexical tokenization, syntactic analysis, semantic analysis, code generation and optimization. In this course, we will learn how these parts work together, and we will practically create a lexical tokenizer and a parser for grammars to solve some tasks (e.g. parsing grammars for mathematical equations or a simple programming language to bytecode that runs on a virtual machine). To understand how this works, we will brush up on some fundamentals, for example in the field of regular and context-free grammars, data structures, finite state machines and push-down automata. The goal is to practically be able to build a real lexer and parser for a (simple) language that can accomplish a specific task.</p>
Lernziele: Fachkompetenz
<p>Nach der erfolgreichen Absolvierung des Teilmoduls sind die Studierenden in der Lage, die Relevanz der Disziplin Compilerbau und die Schnittstellen zu anderen Disziplinen der Informatik zu kennen (1), den Unterschied zwischen den verschiedenen Grammatiken und der Maschinenmodelle, denen sie entsprechen zu kennen, reguläre Ausdrücke formulieren zu können (2), eine Grammatik für bestimmte Anwendungen definieren zu können (2), sowie mit Scanner- und Parsergeneratoren einen vollständigen Parser für ein gegebenes Problem zu erstellen (3). Weiterhin kennen Studierende Grundlagen der Code-Optimierung sowie Maschinencodierung (1).</p>

Lernziele: Persönliche Kompetenz
Nach der erfolgreichen Absolvierung des Teilmoduls sind die Studierenden in der Lage, selbständig ein Problem im Bereich der Lerninhalte zu bearbeiten (3) und dabei die korrekten Werkzeuge sowie einen kreativen Lösungsansatz zu wählen (2).
Lehrmedien
Tafel, Notebook, Beamer, Übungsaufgaben, Codebeispiele
Literatur
"Compilers: Principles, Techniques, and Tools (Second Edition)" (Aho, Lam, Sethi, Ullman)
Weitere Informationen zur Lehrveranstaltung
Qualifications needed: Basic knowledge of Linux and C programming. Die Lehrveranstaltung findet vollständig in englischer Sprache statt.

Die Zahlen in Klammern geben die zu erreichenden Niveaustufen an: 1 - kennen, 2 - können, 3 - verstehen und anwenden