

Modulbezeichnung	Theoretische Informatik 2								
Modulverantwortliche(r)	Prof. Dr. C. Lutz								
Modulart	Pflicht/Wahl <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/>								
Spezialisierungsbereich									
Dauer des Moduls	1 Semester								
Kreditpunkte	6 CP								
Arbeitsaufwand	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Berechnung des Workloads</td> <td style="width: 40%;"></td> </tr> <tr> <td>Präsenz</td> <td style="text-align: right;">56 h</td> </tr> <tr> <td>Übungsbetrieb/Prüfungsvorbereitung</td> <td style="text-align: right;">124 h</td> </tr> <tr> <td>Summe</td> <td style="text-align: right; border-top: 1px solid black;">180 h</td> </tr> </table>	Berechnung des Workloads		Präsenz	56 h	Übungsbetrieb/Prüfungsvorbereitung	124 h	Summe	180 h
Berechnung des Workloads									
Präsenz	56 h								
Übungsbetrieb/Prüfungsvorbereitung	124 h								
Summe	180 h								
Turnus des Moduls	angeboten in jedem SoSe								
Voraussetzung für die Teilnahme	Keine <input checked="" type="checkbox"/> Folgende								
Lehr- und Lernformen	Seminar <input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Tutorium <input checked="" type="checkbox"/> Praktikum <input type="checkbox"/> Projekt <input type="checkbox"/>								
Lernziele	<ul style="list-style-type: none"> • Formale Grundlagen und elementare Fragestellungen der Informatik kennen und die fundamentale Rolle der Theorie in der Informatik verstehen. • Konzepte zur formalen Beschreibung und Analyse von Informatiksystemen kennen. • Beherrschung der grundlegenden Methoden aus den Bereichen der Berechenbarkeit und Komplexität. • Beherrschung elementarer Beweistechniken und Beweise selbst durchführen können. • Probleme analysieren, von spezifischen Gegebenheiten abstrahieren und formale Modelle in mathematischen Definitionen darstellen können. • Algorithmen für diese Probleme kennen und auf neue Problemvarianten anwenden können. • Korrektheit von Algorithmen beweisen und Eigenschaften von Algorithmen analysieren können. • Eigenständig und in Gruppen Lösungsstrategien für formale Problemstellungen entwickeln können und Lösungen verständlich präsentieren. 								

Lerninhalte	<p>1 Berechenbarkeit:</p> <ul style="list-style-type: none"> • Turingmaschinen • Linear beschränkte Automaten • Grammatiken der Typen 0 und 1, Abschlusseigenschaften • LOOP-Programme und WHILE-Programme • Primitiv rekursive Funktionen und -rekursive Funktionen • Unentscheidbarkeit • Unentscheidbare Probleme für Turingmaschinen • Satz von Rice • Postsches Korrespondenzproblem • Äquivalenzproblem kontextfreier Grammatiken • Semi-Entscheidbarkeit und Rekursive Aufzählbarkeit • Universelle Turingmaschinen • Reduktionen
Lerninhalte 2	<p>2 Komplexität:</p> <ul style="list-style-type: none"> • Zeit- und Platzbeschränkte Turingsmaschinen • Komplexitätsklassen P, NP, PSpace, ExpTime • P vs NP-Problem • NP-Vollständigkeit • NP-vollständige Probleme aus verschiedenen Gebieten • Komplemente und coNP • Approximation NP-harter Probleme • Satz von Savitch <p>Lehrveranstaltung(en):</p> <ul style="list-style-type: none"> • 03-IBGT-THI2 Theoretische Informatik 2: Berechenbarkeit und Komplexität
Prüfungsformen	MP, Fachgespräch, Klausur, ggf. mit Bonusprüfung
Literatur	<ul style="list-style-type: none"> • J.E. Hopcroft, R. Motwani, J.D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium 2011 • J.E. Hopcroft, R. Motwani, J.D. Ullman: Introduction to Automata Theory, Languages, and Computation (3rd edition). Pearson Education, 2014 • C. Lutz: Theoretische Informatik, Skript 2. Teil