



# Modulhandbuch Master Informatik

Fakultät für Elektrotechnik, Informatik und Mathematik  
Universität Paderborn

**Version: 8. Juni 2017**

# Inhaltsverzeichnis

<b>1</b>	<b>Beschreibung des Studiengangs Master Informatik</b>	<b>4</b>
1.1	Schema der Modulbeschreibungen . . . . .	5
1.2	Liste der Organisationsformen . . . . .	7
1.3	Liste der Prüfungsformen . . . . .	7
1.4	Liste der nichtkognitive Kompetenzen . . . . .	8
<b>2</b>	<b>Studienrichtungen</b>	<b>10</b>
2.1	Algorithm Design . . . . .	11
2.2	Computer Systems . . . . .	12
2.3	Intelligence and Data . . . . .	13
2.4	Networks and Communication . . . . .	14
2.5	Software Engineering . . . . .	15
<b>3</b>	<b>Module</b>	<b>17</b>
3.1	Wahlpflichtmodul: Adaptive Hardware and Systems . . . . .	18
3.2	Wahlpflichtmodul: Advanced Algorithms . . . . .	21
3.3	Wahlpflichtmodul: Advanced Compiler Construction . . . . .	23
3.4	Wahlpflichtmodul: Advanced Complexity Theory . . . . .	25
3.5	Wahlpflichtmodul: Advanced Computer Architecture . . . . .	28
3.6	Wahlpflichtmodul: Advanced Distributed Algorithms and Data Structures . . . . .	30
3.7	Wahlpflichtmodul: Advanced Software Engineering: Methods, Architectures, Industrial Applications . . . . .	32
3.8	Wahlpflichtmodul: Algorithmic Game Theory . . . . .	34
3.9	Wahlpflichtmodul: Algorithms for Highly Complex Virtual Scenes . . . . .	36
3.10	Wahlpflichtmodul: Algorithms for Synthesis and Optimization of Integrated Circuits . . . . .	39
3.11	Wahlpflichtmodul: Architektur paralleler Rechnersysteme . . . . .	41
3.12	Wahlpflichtmodul: Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies . . . . .	44
3.13	Wahlpflichtmodul: Build It, Break It, Fix It . . . . .	47
3.14	Wahlpflichtmodul: Clustering Algorithms . . . . .	50
3.15	Wahlpflichtmodul: Compiler Construction . . . . .	52
3.16	Wahlpflichtmodul: Computational Geometry . . . . .	54
3.17	Wahlpflichtmodul: Deductive Verification . . . . .	56
3.18	Wahlpflichtmodul: Designing code analyses for large-scale software systems . . . . .	58
3.19	Wahlpflichtmodul: Empiric performance evaluation . . . . .	61
3.20	Wahlpflichtmodul: Foundations of Cryptography . . . . .	63
3.21	Wahlpflichtmodul: Fundamentals of Model-Driven Engineering . . . . .	65
3.22	Wahlpflichtmodul: Future Internet . . . . .	68

3.23	Wahlpflichtmodul: Hardware/Software Codesign . . . . .	70
3.24	Wahlpflichtmodul: High-Performance Computing . . . . .	72
3.25	Wahlpflichtmodul: Intelligence in Embedded Systems . . . . .	74
3.26	Wahlpflichtmodul: Interactive Data Visualization . . . . .	76
3.27	Wahlpflichtmodul: Kontextuelle Informatik . . . . .	79
3.28	Wahlpflichtmodul: Language-Based Security . . . . .	82
3.29	Wahlpflichtmodul: Linear and Integer Optimization . . . . .	84
3.30	Wahlpflichtmodul: Logic and Automated Reasoning . . . . .	86
3.31	Wahlpflichtmodul: Logic Programming for Artificial Intelligence . . . . .	89
3.32	Wahlpflichtmodul: Machine Learning I . . . . .	92
3.33	Wahlpflichtmodul: Machine Learning II . . . . .	94
3.34	Pflichtmodul: Master-Abschlussarbeit . . . . .	96
3.35	Wahlpflichtmodul: Mobile Communication . . . . .	99
3.36	Wahlpflichtmodul: Model Checking . . . . .	101
3.37	Wahlpflichtmodul: Model-Based Interface Development . . . . .	103
3.38	Wahlpflichtmodul: Network Simulation . . . . .	106
3.39	Wahlpflichtmodul: Networked Embedded Systems . . . . .	108
3.40	Wahlpflichtmodul: Planning and Heuristic Search . . . . .	110
3.41	Pflichtmodul: Projektgruppe . . . . .	113
3.42	Wahlpflichtmodul: Public-Key Cryptography . . . . .	116
3.43	Wahlpflichtmodul: Reconfigurable Computing . . . . .	118
3.44	Wahlpflichtmodul: Routing and Data Management in Networks . . . . .	120
3.45	Pflichtmodul: Seminar I . . . . .	122
3.46	Pflichtmodul: Seminar II . . . . .	124
3.47	Wahlpflichtmodul: Software Analysis . . . . .	126
3.48	Wahlpflichtmodul: Software Quality Assurance . . . . .	128
3.49	Pflichtmodul: Studium Generale . . . . .	130
3.50	Wahlpflichtmodul: Type Systems for Correctness and Security . . . . .	132
3.51	Wahlpflichtmodul: Usability Engineering Practice . . . . .	135
3.52	Wahlpflichtmodul: Vehicular Networking . . . . .	138
3.53	Wahlpflichtmodul: VLSI Testing . . . . .	140
<b>A</b>	<b>Überblickstabellen</b>	<b>142</b>
A.1	Studienrichtungen und Module . . . . .	143
A.2	Module und Lehrveranstaltungen . . . . .	145

# Kapitel 1

## Beschreibung des Studiengangs Master Informatik

Ziel des viersemestrigen Master-Studienganges Informatik ist die Vertiefung der in einem Bachelor-Studiengang erworbenen Kenntnisse in einem oder mehreren Gebieten der Informatik. Dabei ist eines von fünf Gebieten als Vertiefungsgebiet (Focus Area) zu wählen - in diesem Gebiet müssen mindestens 3 Module (je 6 ECTS) studiert werden; außerdem muss die Masterarbeit in diesem Gebiet geschrieben werden. In mindestens einem weiteren Gebiet ist ein Modul zu absolvieren. Hinzu kommt eine Projektgruppe (20 ECTS), eine zweisemestrige Veranstaltungsform, in der im Team an einem forschungsnahen Thema gearbeitet wird. Im Übrigen können Studierenden ihr Studium frei organisieren.

## 1.1 Schema der Modulbeschreibungen

Die Modulbeschreibungen sind nach folgendem Schema einheitlich strukturiert:

Modulname	<Name des Moduls>
Workload	<Gesamtaufwand in Stunden (Workload ECTS)>
Leistungspunkte	<Gesamtaufwand in Leistungspunkten ECTS>
Studiensemester	<Liste der Lehrveranstaltungen in diesem Modul mit Zielsemester>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
<Liste der im Modul enthaltenen Lehrveranstaltungen mit Aufteilung des Workloads in Kontaktzeit und Selbststudium, Sprache in der die Veranstaltung gehalten wird, Winter- oder Sommersemester und ungefähre Gruppengröße.>	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
<Liste der im Modul enthaltenen Wahlmöglichkeiten.>	
<b>Teilnahmevoraussetzungen</b>	
<Voraussetzungen für die Teilnahme am Modul.>	
<b>Empfohlene Kenntnisse</b>	
<Die Angaben sind als Empfehlungen zu verstehen, nicht jedoch als zu überprüfende Voraussetzungen.>	
<b>Inhalte</b>	
<Aufzählung der wesentlichen Inhalte der enthaltenen Lehrveranstaltungen.>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<Aufzählung der erreichten Kenntnisse, Fähigkeiten und Fachkompetenzen.>	
<b>Nichtkognitive Kompetenzen</b>	
<Zusammenfassung aller nichtkognitiver Kompetenzen, die in den Lehrveranstaltungen des Moduls vermittelt werden.>	
<b>Methodische Umsetzung</b>	
<Angaben zu Sozialformen und didaktisch-methodischen Arbeitsweisen in den Veranstaltungen.>	
<b>Prüfungsleistung (Dauer)</b>	
<Form in Dauer der im Modul zu erbringenden Prüfungsleistung.>	
<b>Modulteilprüfungen</b>	
<Form der im Modul zu erbringenden Modulteilprüfung.>	

<b>Studienleistung / qualifizierte Teilnahme</b>
<Form der im Modul zu erbringenden Studienleistungen oder qualifizierter Teilnahmen.>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
<Formale Voraussetzungen für Teilnahme an der Modulprüfung.>
<b>Voraussetzungen für die Vergabe von Credits</b>
<Formale Voraussetzungen für die Vergabe von Credits.>
<b>Gewichtung für die Gesamtnote</b>
<Gesamtgewichtung des Moduls bei der Berechnung des Notendurchschnitts.>
<b>Modul wird in folgenden Studiengängen verwendet</b>
<Liste der Studiengänge, in denen dieses Modul verwendet wird.>
<b>Modulbeauftragte/r</b>
<Verantwortlicher für das Modul.>
<b>Lernmaterialien, Literaturangaben</b>
<Angaben zu Literatur, Vorlesungsskripten, etc.>
<b>Sonstige Hinweise</b>
<Sonstige Hinweise.>

## 1.2 Liste der Organisationsformen

Die folgenden Organisationsformen werden in diesem Studiengang verwendet:

### **Abschlussarbeit**

**Praktikum** In Kleingruppen arbeiten Studierende an praktischen Aufgaben.

### **Seminar**

**Vorlesung mit Übung** Eine Kombination aus Vorlesung und begleitenden Übungen, häufig mit praktischen Anteilen und Hausaufgaben.

**Vorlesung mit Übung und Praktikum** Eine Vorlesung mit Übungen wird mit einem Praktikumsteil kombiniert.

**Vorlesungen, Seminare, Projekte** Eine beliebige Kombination von Veranstaltungen für das Studium Generale.

## 1.3 Liste der Prüfungsformen

Die folgenden Organisationsformen werden in diesem Studiengang verwendet:

**Abschlussarbeit** Eine Abschlussarbeit soll zeigen, dass die Kandidatin oder der Kandidat die Fähigkeit besitzt, innerhalb einer bestimmten Frist ein Problem der Informatik auf der Grundlage wissenschaftlicher Methoden zu bearbeiten.

**Mündliche Prüfung** In den mündlichen Prüfungen soll die Kandidatin bzw. der Kandidat nachweisen, dass sie bzw. er die Zusammenhänge des Prüfungsgebietes erkennt und spezielle Fragestellungen in diese Zusammenhänge einzuordnen vermag. Durch die mündlichen Prüfungen soll ferner festgestellt werden, ob die Kandidatin bzw. der Kandidat über breites Grundlagenwissen verfügt. Mündliche Prüfungen dauern in der Regel mindestens 25 und höchstens 50 Minuten. Bei Gruppenprüfungen verlängert sich die Gesamtprüfungsdauer entsprechend der Kandidatenzahl.

**Phasenbezogene Prüfung (100 % der Modulnote)** In einer phasenbezogenen Prüfung ist die erfolgreiche Bearbeitung von Projekten durch die Abgabe von Software und Dokumentation nachzuweisen. Es wird eine Note für die Gesamtheit der bearbeiteten Projekte vergeben.

**Prüfung im Studium Generale** In einer der im Studium Generale gewählten Veranstaltungen wird eine mündliche oder schriftliche Prüfungsleistung erbracht. Dabei handelt es sich in der Regel um eine Klausur (maximal vier Stunden), eine Hausarbeit (maximal 25 Seiten) oder eine mündliche Prüfung (maximal 45 Minuten).

**Seminarvortrag (45-60 Minuten) und schriftliche Ausarbeitung** In einem Seminarvortrag sollen die Studierenden nachweisen, dass sie zur wissenschaftlichen Ausarbeitung eines Themas in der Lage sind und die Ergebnisse vortragen können. In der schriftlichen Ausarbeitung sollen sie zeigen, dass sie in der Lage sind Ergebnisse gemäß wissenschaftlichen Standards darzustellen.

**Studienleistung** Als Studienleistung können Übungsaufgaben verlangt werden, die in der Regel wöchentlich als Hausaufgaben und/oder Präsenzaufgaben gestellt werden. Studienleistungen werden mit bestanden oder nicht bestanden bewertet. Bestanden Studienleistungen sind Voraussetzung zur Teilnahme an einer Modulabschlussprüfung.

**Qualifizierte Teilnahme** Die qualifizierte Teilnahme wird in der Regel durch eine Praktikumsarbeit mit anschließendem Gespräch nachgewiesen. Eine qualifizierte Teilnahme liegt dann vor, wenn die erbrachten Leistungen erkennen lassen, dass eine mehr als nur oberflächliche Beschäftigung mit den Gegenständen, die einer Aufgabenstellung zugrunde lagen, stattgefunden hat. Qualifizierte Teilnahmen sind zum Bestehen eines Moduls nachzuweisen.

**Qualifizierte Teilnahme im Studium Generale** In den Veranstaltungen des Studium Generale, bei denen keine Prüfung im Studium Generale abgelegt wird, ist die qualifizierte Teilnahme nachzuweisen.

## 1.4 Liste der nichtkognitive Kompetenzen

Dieser Studiengang baut die folgenden nichtkognitive Kompetenzen auf:

### **Einsatz und Engagement**

- Gefühl der Verpflichtung informatorische Aufträge zu erfüllen
- Durchhaltevermögen bei der Bearbeitung informatischer Aufträge

### **Empathie**

- Fähigkeit zum Perspektiv- und Rollenwechsel
- Fähigkeit sich in informatikfremde Personen hineinzuversetzen
- Erkennen der Anliegen informatikfremder Personen

### **Gruppenarbeit**

Die Fähigkeit, effektiv und effizient in Gruppen bis zu mittlerer Größe (ca. 15 Personen) zu arbeiten.

### **Haltung und Einstellung**

- Affinität gegenüber informatischen Problemen
- Bereitschaft sich informatischen Herausforderungen zu stellen
- Sozial-kommunikative Fähigkeiten als bedeutsam beurteilen

### **Kooperationskompetenz**

- Hilfs- und Kooperationsbereitschaft
- Sprachkompetenz
- Kommunikative Fähigkeiten
- Diskussionsbereitschaft gegenüber informatischen Themen
- Informatische Themen präsentieren können
- Fähigkeit und Bereitschaft informatisches Wissen weiterzugeben
- Fähigkeit und Bereitschaft zu konstruktiver Kritik
- Fähigkeit und Bereitschaft Absprachen zu treffen und einzuhalten
- Bereitschaft entlang der Absprachen zu handeln
- Bereitschaft fremde Ideen anzunehmen

### **Lernkompetenz**



- Fähigkeit und Bereitschaft zu lebenslangem Lernen
- Fähigkeit und Bereitschaft zu problemorientiertem Lernen
- Fähigkeit und Bereitschaft kooperativem Lernen
- Fähigkeit zur Selbstorganisation von Lernprozessen und zu selbstständigem Lernen

### **Lernmotivation**

- Bereitschaft informatische Fähigkeiten und informatorisches Wissen zu erweitern
- Bereitschaft informatische Aufträge zu erfüllen

### **Medienkompetenz**

- Nutzung problemorientierter Lern- und Entwicklungsumgebungen
- Nutzung von Werkzeugen zum wissenschaftlichen Schreiben
- Nutzung von Werkzeugen zum Präsentieren wissenschaftlicher Resultate

### **Motivationale und volitionale Fähigkeiten**

- Offenheit neuen Ideen und Anforderungen gegenüber
- Bereitschaft neue und unvertraute Lösungswege anzuwenden
- Kritikfähigkeit gegenüber einem und reflektierten Umgang mit rezeptartigen Lösungswegen

### **Schreib- und Lesekompetenz (wissenschaftlich)**

- Fähigkeit Quellen zu recherchieren und reflektiert zu beurteilen
- Fähigkeit informatische Sachverhalte sinnvoll zu strukturieren
- Fähigkeit eigene Ideen von anderen korrekt abzugrenzen (Vermeidung von Plagiaten)

### **Selbststeuerungskompetenz**

- Verbindlichkeit
- Disziplin
- Termintreue
- Kompromissbereitschaft
- Übernahme von Verantwortung
- Geduld
- Selbstkontrolle
- Gewissenhaftigkeit
- Zielorientierung
- Motivation
- Aufmerksamkeit

## Kapitel 2

# Studienrichtungen

## 2.1 Algorithm Design

Studienrichtung	Algorithm Design
Koordination	Prof. Dr. rer. nat. Johannes Blömer Codes und Kryptographie Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Advanced Algorithms (S. 21)</li> <li>• Advanced Complexity Theory (S. 25)</li> <li>• Advanced Distributed Algorithms and Data Structures (S. 30)</li> <li>• Algorithmic Game Theory (S. 34)</li> <li>• Algorithms for Highly Complex Virtual Scenes (S. 36)</li> <li>• Clustering Algorithms (S. 50)</li> <li>• Computational Geometry (S. 54)</li> <li>• Foundations of Cryptography (S. 63)</li> <li>• Linear and Integer Optimization (S. 84)</li> <li>• Public-Key Cryptography (S. 116)</li> <li>• Routing and Data Management in Networks (S. 120)</li> </ul>
Beschreibung	
<p>In diesem Vertiefungsgebiet konzentrieren sich Studierende auf</p> <ul style="list-style-type: none"> <li>• wesentliche Techniken zum Entwurf effizienter Algorithmen</li> <li>• Anwendungsgebiete des Entwurfs effizienter Algorithmen, z.B. Computergrafik, Netzwerke, Big Data, ...</li> <li>• Grenzen für den Entwurf effizienter Algorithmen, d.h. Komplexitätstheorie</li> <li>• konstruktiver Anwendungen der Grenzen des Entwurfs effizienter Algorithmen in Kryptographie und IT-Sicherheit</li> </ul>	

## 2.2 Computer Systems

Studienrichtung	Computer Systems
Koordination	Prof. Dr. Marco Platzner Technische Informatik Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Adaptive Hardware and Systems (S. 18)</li> <li>• Advanced Computer Architecture (S. 28)</li> <li>• Algorithms for Synthesis and Optimization of Integrated Circuits (S. 39)</li> <li>• Architektur paralleler Rechnersysteme (S. 41)</li> <li>• Empiric performance evaluation (S. 61)</li> <li>• Hardware/Software Codesign (S. 70)</li> <li>• High-Performance Computing (S. 72)</li> <li>• Intelligence in Embedded Systems (S. 74)</li> <li>• Reconfigurable Computing (S. 118)</li> <li>• VLSI Testing (S. 140)</li> </ul>
Beschreibung	
<p>Das Vertiefungsgebiet "Computersysteme" behandelt vertiefend und im technischen Detail verschiedene Aspekte von modernen Computersystemen. Im Vordergrund stehen dabei die Analyse und Bewertung von Rechnerarchitekturen, systematische Methoden für den Entwurf und die Optimierung von Computersystemen, insbesondere das Zusammenspiel von Hardware und Software, sowie Programmiermodelle und -methoden für die stark an Bedeutung gewinnenden parallelen und spezialisierten Rechnerarchitekturen.</p>	

## 2.3 Intelligence and Data

Studienrichtung	Intelligence and Data
Koordination	Prof. Dr. Eyke Hüllermeier Intelligente Systeme Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Clustering Algorithms (S. 50)</li> <li>• Intelligence in Embedded Systems (S. 74)</li> <li>• Interactive Data Visualization (S. 76)</li> <li>• Logic Programming for Artificial Intelligence (S. 89)</li> <li>• Logic and Automated Reasoning (S. 86)</li> <li>• Machine Learning I (S. 92)</li> <li>• Machine Learning II (S. 94)</li> <li>• Planning and Heuristic Search (S. 110)</li> </ul>
Beschreibung	
	<p>Intelligente Systeme sind Computersysteme, deren Verhalten durch Methoden und Algorithmen der Künstlichen Intelligenz (KI) gesteuert wird. Solche Systeme gewinnen kontinuierlich an Bedeutung, nicht nur auf wissenschaftlicher Ebene innerhalb der Informatik, sondern auch im sozialen und gesellschaftlichen Kontext: Autonome oder teilautonome Systeme wie Serviceroboter, selbstfahrende PKWs oder medizinische Diagnosesysteme werden unser privates und berufliches Leben in absehbarer Zukunft tiefgreifend verändern. Neben methodischen Fortschritten und einer Steigerung der Rechenleistung durch schnellere Hardware ist die rasante Entwicklung von KI-Systemen in der letzten Dekade vor allem einer Datenexplosion zu verdanken: Die Verfügbarkeit großer Mengen von Daten oder sensorisch erfasster Beobachtungen aus ihrer Umgebung versetzt intelligente Systeme in die Lage, ihr Verhalten durch Adaption und Lernen selbständige zu optimieren.</p> <p>Dieses Modul greift wichtige Aspekte des Entwurfs intelligenter Systeme auf und vermittelt entsprechende theoretische und methodische Grundlagen. Die Inhalte des Moduls erstrecken sich von Themen wie Maschinelles Lernen und Datenanalyse über das Datenmanagement bis hin zur graphischen Datenverarbeitung und Anwendungen in der Robotik und Schwarmintelligenz.</p>

## 2.4 Networks and Communication

Studienrichtung	Networks and Communication
Koordination	Prof. Dr. rer. nat. Holger Karl Rechnernetze Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Advanced Distributed Algorithms and Data Structures (S. 30)</li> <li>• Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 44)</li> <li>• Empiric performance evaluation (S. 61)</li> <li>• Future Internet (S. 68)</li> <li>• Mobile Communication (S. 99)</li> <li>• Network Simulation (S. 106)</li> <li>• Networked Embedded Systems (S. 108)</li> <li>• Routing and Data Management in Networks (S. 120)</li> <li>• Vehicular Networking (S. 138)</li> </ul>
Beschreibung	<p>Das Vertiefungsgebiet “Netze und Kommunikation” lehrt Architekturen, Methoden und Systeme moderner Kommunikationstechnik. Hierzu werden Methoden unterschiedlicher Abstraktionsebenen untersucht, beginnend bei der physikalischen Übertragung bis hin zum Anwendungsentwurf in verteilten Umgebungen. Dabei werden unterschiedliche Systemklassen behandelt, von klassischer Mobilkommunikation über ad hoc Netze und Fahrzeugkommunikation bis zur Vernetzung in Rechenzentren und Architekturen des zukünftigen Internets. Dabei wird die Brücke zu verteilten System hergestellt. Neben Fragen des Architekturentwurfs, der Methoden- und Protokollgestaltung steht dabei stets die Frage der Bewertung solcher Verfahren im Raum; hierzu werden experimentelle und statistische Verfahren besprochen.</p>

## 2.5 Software Engineering

Studienrichtung	Software Engineering
Koordination	Prof. Dr. Gregor Engels Datenbank- und Informationssysteme Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Advanced Compiler Construction (S. 23)</li> <li>• Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 32)</li> <li>• Build It, Break It, Fix It (S. 47)</li> <li>• Compiler Construction (S. 52)</li> <li>• Deductive Verification (S. 56)</li> <li>• Designing code analyses for large-scale software systems (S. 58)</li> <li>• Empiric performance evaluation (S. 61)</li> <li>• Fundamentals of Model-Driven Engineering (S. 65)</li> <li>• High-Performance Computing (S. 72)</li> <li>• Kontextuelle Informatik (S. 79)</li> <li>• Language-Based Security (S. 82)</li> <li>• Logic Programming for Artificial Intelligence (S. 89)</li> <li>• Model Checking (S. 101)</li> <li>• Model-Based Interface Development (S. 103)</li> <li>• Software Analysis (S. 126)</li> <li>• Software Quality Assurance (S. 128)</li> <li>• Type Systems for Correctness and Security (S. 132)</li> <li>• Usability Engineering Practice (S. 135)</li> </ul>

**Beschreibung**

In dieser Studienrichtung können die Studierenden Konzepte, Sprachen, Methoden, Techniken und Werkzeuge für eine systematische Entwicklung von Softwaresysteme erlernen. Dies umfasst

- konstruktive Techniken zur Realisierung von funktionalen und nicht-funktionalen Aspekten eines Softwaresystems,
- formale and informelle analytische Techniken, um eine hohe Qualität eines Systems zu erzielen und
- systematische Techniken, um situationspezifische Vorgehensmodelle zu definieren.



# Kapitel 3

# Module

### 3.1 Wahlpflichtmodul: Adaptive Hardware and Systems

Modulname	Adaptive Hardware and Systems / Adaptive Hardware and Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Adaptive Hardware and Systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Adaptive Hardware and Systems: Vorlesung ( 45h / 105h / EN / SS / 25 ) Adaptive Hardware and Systems: Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Adaptive Hardware and Systems: Grundlegende Kenntnisse in Digitaltechnik, Rechnerarchitektur, Programmierkenntnisse	
<b>Inhalte</b>	
<p>Adaptive Hardware and Systems: Adaptation bezeichnet die Fähigkeit eines Systems, die eigene Funktions- und Leistungsfähigkeit an sich verändernde Bedingungen anzupassen und aufrechtzuerhalten. Die Vorlesung konzentriert sich auf adaptive Hardwaresysteme. Nach einer kurzen Einführung rekonfigurierbarer Bausteine, behandelt die Vorlesung die Prinzipien der Computational Intelligence für die Umsetzung der Adaptations- und Optimierungsmechanismen.</p> <p>In Laborübungen werden mitunter folgende Programmieraufgaben gelöst: lernende Systeme (adaptive Signalklassifikation), Optimierung Chipentwurf (Floorplanning, Placement) und parallele Optimierung von Prozessorcaches (PC2 Cluster).</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Nach Abschluss des Moduls können die Studierenden</p> <ul style="list-style-type: none"> <li>die Prinzipien moderner Metaheuristiken erläutern und anwenden,</li> <li>Optimierungsaufgaben typisieren und formal modellieren,</li> <li>Zielfunktionen und Randbedingungen definieren, Lösungsansätze entwickeln und</li> <li>wesentliche Herausforderungen beim automatisierten Entwurf, Optimierung und Adaption digitaler Schaltungen und moderner Computersysteme benennen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Lernkompetenz</li> <li>Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	

<b>Methodische Umsetzung</b>
Adaptive Hardware and Systems: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Selbststudium und Diskussion wissenschaftlicher Publikationen</li> <li>• Praktische Programmierprojekte</li> <li>• Lösung paralleler Optimierungsaufgaben am PC2 Cluster</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Paul Kaufmann
<b>Lernmaterialien, Literaturangaben</b>
Adaptive Hardware and Systems: Folien; ausgewählte Fachartikel; Lehrbücher <ul style="list-style-type: none"> <li>• Weicker, Karsten, "Evolutionäre Algorithmen", Springer, 2007. ISBN 978-3-8351-9203-4</li> <li>• Kruse et al.: "Computational Intelligence - A Methodological Introduction", Springer, 2013. ISBN 978-1-4471-5012-1</li> <li>• Kruse et al.: "Computational Intelligence [DE]", Vieweg+Teubner-Verlag, Wiesbaden, 2011. ISBN 978-3-8348-1275-9</li> <li>• Wang et al.: "Électronic Design Automation", Morgan Kaufmann, 2009. ISBN: 0-1237-4364-8</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.2 Wahlpflichtmodul: Advanced Algorithms

Modulname	Advanced Algorithms / Advanced Algorithms
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Advanced Algorithms : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Algorithms: Vorlesung ( 45h / 105h / EN / WS / 50 ) Advanced Algorithms: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Algorithms: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen werden vorausgesetzt.	
<b>Inhalte</b>	
Advanced Algorithms: Dieser Kurs präsentiert fortgeschrittenen Algorithmen und algorithmische Paradigmen für grundlegenden Probleme. Insbesondere werden dabei Methoden wie Randomisierung und Derandomisierung, sowie die Konzepte von Approximations- und Onlinealgorithmen anhand wichtiger algorithmischer Probleme vorgestellt. In allen Fällen werden Korrektheitsbeweise und Laufzeitanalysen durchgeführt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden wenden fortgeschrittenen algorithmische Entwurfsmethoden wie Randomisierung, Approximation und Onlinealgorithmen auf neue Probleme und analysieren sie unter Nutzung von kombinatorischen und wahrscheinlichkeitstheoretischen Methoden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Algorithms: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb.</li> <li>• Übungen in Kleingruppen.</li> <li>• erwartete Aktivitäten der Studierenden: aktive Mitarbeit bei Präsenzübungen, Hausaufgaben.</li> </ul>	

<ul style="list-style-type: none"> <li>• Übungsblätter, Lösungen werden in Übungsgruppen vorgestellt und diskutiert.</li> <li>• In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Friedhelm Meyer auf der Heide
<b>Lernmaterialien, Literaturangaben</b>
Advanced Algorithms: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter
<b>Sonstige Hinweise</b>
keine

### 3.3 Wahlpflichtmodul: Advanced Compiler Construction

Modulname	Advanced Compiler Construction / Advanced Compiler Construction
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Compiler Construction : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Compiler Construction: Vorlesung ( 45h / 105h / EN / WS / 25 ) Advanced Compiler Construction: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Compiler Construction: Grundlegende Kenntnisse aus dem Compilerbau werden vorausgesetzt.	
<b>Inhalte</b>	
Advanced Compiler Construction: Ziel des Kurses ist es, fortgeschrittene Techniken des Compilerbaus (u.a. auch aus dem Security Bereich, siehe LVA "Sprach-basierte Sicherheit") zu vermitteln.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Den Studenten erwerben fundierte Kenntnisse aus dem Bereich der Just-in-Time Compiler. Probleme dynamischer Programmiersprachen werden eingehend erläutert und verschiedene Lösungen werden anhand konkreter Implementierungsbeispiele veranschaulicht.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Lernkompetenz</li> <li>Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Compiler Construction: <ul style="list-style-type: none"> <li>Vorlesung mit Beamer und Tafelanschrieb.</li> <li>Begleitende Implementierung als Übungsteil.</li> </ul>	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof.Dr. Stefan Brunthaler
<b>Lernmaterialien, Literaturangaben</b>
Advanced Compiler Construction: keine
<b>Sonstige Hinweise</b>
keine



### 3.4 Wahlpflichtmodul: Advanced Complexity Theory

Modulname	Advanced Complexity Theory / Advanced Complexity Theory
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Complexity Theory : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Complexity Theory: Vorlesung ( 45h / 105h / EN / WS oder SS / 25 ) Advanced Complexity Theory: Übung ( 30h / 0h / EN / WS oder SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Complexity Theory: Grundlagen über Komplexitätstheorie (u.a. Turingmaschinen, NP-Vollständigkeit)	
<b>Inhalte</b>	
<p>Advanced Complexity Theory: Komplexitätstheorie beschäftigt sich mit der Bestimmung der Größe von Ressourcen (z.B. Laufzeit, Speicherverbrauch), die notwendig und hinreichend ist für die Lösung eines bestimmten algorithmischen Problems (z.B. Problem des Handlungsreisenden (TSP)) auf einem bestimmten Computermodell (z.B. Turing-Maschine). Ein Ansatz ist die Komplexitätsklassen wie z.B. P, NP, PSPACE zu definieren, um die Problemkomplexität mit Hilfe der Vollständigkeit in einer solchen Klasse zu klassifizieren, wie z.B. die berühmte Klasse der NP-vollständigen Probleme. Dies ergibt bedingte Aussagen wie „Wenn NP nicht gleich P, dann ist TSP nicht in Polynomialzeit lösbar“. Dieser Zweig der Komplexitätstheorie wird oft als strukturelle Komplexitätstheorie bezeichnet. Im Gegensatz dazu ist das Beweisen expliziter Untergrenzen für bestimmte Probleme das Thema der so genannten konkreten Komplexitätstheorie. Da niemand derzeit in der Lage ist superlineare Zeitschranken für explizit definierte Probleme in allgemeinen Rechenmodellen wie Turingmaschinen zu beweisen, betrachtet man etwas eingeschränkt Modelle wie 1-Band Turingmaschinen, monotone Bool'sche Schaltkreise, Bool'sche Schaltkreise mit beschränkter Tiefe, algebraische Berechnungsmodelle und verschiedene Arten von parallelen Berechnungsmodellen. Die Vorlesung gibt eine Übersicht von Ansätzen um solche unteren Schranke in verschiedenen Modellen zu beweisen.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Die Studierenden beherrschen die wichtigsten Techniken im Bereich der Komplexitätstheorie wie Reduktionen, Diagonalisierung, Randomisierung und Relativierung. Sie können entscheiden, in welche Komplexitätsklassen sich der Speicherplatzbedarf und die Laufzeitanforderungen von algorithmischen Problemen einordnen lassen. Sie können mit Hilfe der Konzepte der Komplexitätstheorie Hypothesen aufstellen und diese falsifizieren oder verifizieren. Sie können die Zusammenhänge zwischen Komplex-</p>	

itätsklassen wie P und NL erläutern.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Advanced Complexity Theory:</p> <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Mündliche Prüfung (ca. 40 Minuten)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: schriftliche Übungsaufgaben  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Friedhelm Meyer auf der Heide
<b>Lernmaterialien, Literaturangaben</b>
<p>Advanced Complexity Theory: C.H. Papadimitriou, Computational Complexity, Addison-Wesley, S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press; Foliensatz der Vorlesung, Übungsblätter</p>

<b>Sonstige Hinweise</b>
keine

### 3.5 Wahlpflichtmodul: Advanced Computer Architecture

Modulname	Advanced Computer Architecture / Advanced Computer Architecture
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Advanced Computer Architecture : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Computer Architecture: Vorlesung ( 45h / 105h / EN / WS / 50 ) Advanced Computer Architecture: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Computer Architecture: Grundlegende Kenntnisse in Rechnerarchitektur.	
<b>Inhalte</b>	
Advanced Computer Architecture: Die Lehrveranstaltung vermittelt die wesentliche Konzepte und Methoden, die beim Entwurf moderner Prozessoren Verwendung finden. Es werden Ansätze zur Nutzung von Parallelität auf der Instruktions-, Daten- und Thread-Ebene besprochen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• die Prinzipien modernen Speicherhierarchien zu erklären,</li> <li>• die verschiedenen Ebenen der Parallelität zu analysieren,</li> <li>• die Eignung unterschiedlicher Architekturkonzepte einzuschätzen und dadurch</li> <li>• moderne Entwicklungen der Rechnerarchitektur zu bewerten.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Computer Architecture: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafel</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Rechnerübungen mit Simulationswerkzeugen</li> <li>• Analyse von Fallbeispielen</li> </ul>	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Advanced Computer Architecture: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• Hennessey, Patterson: Computer Architecture: A Quantitative Approach (5th edition), Morgan Kaufmann, 2012.</li> <li>• Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.6 Wahlpflichtmodul: Advanced Distributed Algorithms and Data Structures

Modulname	Advanced Distributed Algorithms and Data Structures / Advanced Distributed Algorithms and Data Structures
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Advanced Distributed Algorithms and Data Structures : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Distributed Algorithms and Data Structures: Vorlesung ( 45h / 105h / EN / WS / 30 ) Advanced Distributed Algorithms and Data Structures: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Distributed Algorithms and Data Structures: Algorithmen und Datenstrukturen, verteilte Algorithmen und Datenstrukturen	
<b>Inhalte</b>	
Advanced Distributed Algorithms and Data Structures: Die Vorlesung stellt fortgeschrittene Methoden vor, um hochskalierbare verteilte Algorithmen und Datenstrukturen zu entwickeln. Die Vorlesung teilt sich dabei in separate Bereiche auf, die aktuell relevant für den Bereich der verteilten Systeme sind. Dazu gehören lokalitätserhaltende Systeme, robust Informationssysteme, und programmierbare Materie.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende lernen fortgeschrittene Methoden und Verfahren für aktuell sehr relevante verteilte Systeme kennen. Sie können Verfahren an neue Situationen anpassen und deren Komplexität bestimmen. Sie können grundlegende Verfahren implementieren.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Distributed Algorithms and Data Structures: Vorlesung mit Übungen und Softwareprojekt	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Christian Scheideler
<b>Lernmaterialien, Literaturangaben</b>
Advanced Distributed Algorithms and Data Structures: Skript
<b>Sonstige Hinweise</b>
keine

### 3.7 Wahlpflichtmodul: Advanced Software Engineering: Methods, Architectures, Industrial Applications

Modulname	Advanced Software Engineering: Methods, Architectures, Industrial Applications / Advanced Software Engineering: Methods, Architectures, Industrial Applications
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Software Engineering: Methods, Architectures, Industrial Applications : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Vorlesung ( 45h / 105h / EN / WS / 92 ) Advanced Software Engineering: Methods, Architectures, Industrial Applications: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Modellbasierte Softwareentwicklung	
<b>Inhalte</b>	
Advanced Software Engineering: Methods, Architectures, Industrial Applications: In der Veranstaltung werden fortgeschrittene Konzepte und Methoden der Softwaretechnik vermittelt. Hierzu gehören Ansätze zur Entwicklung von Architekturen, moderne Softwareentwicklungsmethoden sowie Elemente des Projektmanagements. Im Einzelnen werden die folgenden Themen behandelt: - Architekturen (serviceorientierte Architekturen, Microservices, Web Services, Architekturen für mobile Anwendungen, Architekturframeworks, Quasar Enterprise) - Methoden (Agile Entwicklungsmethoden wie Scrum, Situational Method Engineering, Requirements Engineering, DevOps) - Projektmanagement (Aufwandsschätzung, Ökonomie von Softwareprojekten) Die Veranstaltung beinhaltet auch Vorträge von erfahrenen Experten aus der Industrie, die über den Einsatz der modernen Software-Engineering-Techniken in der Praxis berichten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage, wesentliche Konzepte einer modernen Softwareentwicklung zu benennen und einzusetzen. Sie kennen die wesentlichen Techniken, um Architekturen von Softwaresystemen zu entwickeln, Softwareentwicklungsmethoden situationsgerecht anzupassen und einzusetzen sowie Pro-	



jektmanagementaufgaben wie z.B. Aufwandsschätzung durchzuführen.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Empathie</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. In praktischen Übungen werden die erlernten Kenntnisse angewendet.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gregor Engels
<b>Lernmaterialien, Literaturangaben</b>
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.8 Wahlpflichtmodul: Algorithmic Game Theory

Modulname	Algorithmic Game Theory / Algorithmic Game Theory
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Algorithmic Game Theory : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithmic Game Theory: Vorlesung ( 45h / 105h / EN / WS oder SS / 25 ) Algorithmic Game Theory: Übung ( 30h / 0h / EN / WS oder SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithmic Game Theory: Algorithmen und Datenstrukturen, Lineare Algebra, Wahrscheinlichkeitstheorie	
<b>Inhalte</b>	
Algorithmic Game Theory: In dieser Vorlesung werden die wichtigsten Ergebnisse und Methoden aus dem Gebiet der Algorithmischen Spieltheorie vorgestellt. Es werden zahlreiche Fragestellungen der nicht kooperativen Spieltheorie sowie aus dem Bereich des Social Choice behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende lernen die wesentlichen Konzepte der kooperativen und nicht-kooperativen Spieltheorie kennen, insbesondere Standard Modelle und Lösungskonzepte. Sie verstehen, eine Vielzahl von Algorithmischentechniken und Komplexitätsresultate zur Berechnung spieltheoretische Lösungskonzepte. Sie können Lösungskonzepte, Algorithmen und Komplexitätsresultate auf neue Spiele, die Varianten bekannter Beispiele sind, anwenden. Sie verstehen den Stand der algorithmischen Forschung in einigen Bereichen, einschließlich neuer Trends und offene Probleme.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Algorithmic Game Theory:	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Jun.-Prof. Dr. Alexander Skopalik
<b>Lernmaterialien, Literaturangaben</b>
Algorithmic Game Theory: Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. 2007. Algorithmic Game Theory. Cambridge University Press, New York, NY, USA. Slides Exercises
<b>Sonstige Hinweise</b>
keine

### 3.9 Wahlpflichtmodul: Algorithms for Highly Complex Virtual Scenes

Modulname	Algorithms for Highly Complex Virtual Scenes / Algorithms for Highly Complex Virtual Scenes
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Algorithmen für hochkomplexe virtuelle Szenen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithmen für hochkomplexe virtuelle Szenen: Vorlesung ( 45h / 105h / EN / WS / 40 ) Algorithmen für hochkomplexe virtuelle Szenen: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithmen für hochkomplexe virtuelle Szenen: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen werden angenommen.	
<b>Inhalte</b>	
Algorithmen für hochkomplexe virtuelle Szenen: Walkthrough-Systeme erlauben das Betrachten und Durchlaufen von virtuellen 3D-Szenen und finden Anwendung in Architekturprogrammen, Simulationen, oder Spielen. Die Effizienz von Echtzeit-Rendering Algorithmen ist entscheidend für eine flüssige und schnelle Darstellung der virtuellen 3D-Szenen in einem Walkthrough-System. Es gibt verschiedene algorithmische Ansätze, um hochkomplexe geometrische 3D-Daten zu reduzieren und eine Darstellung der Daten in Echtzeit zu erreichen. In der Vorlesung werden algorithmische Ansätze aus den Bereichen Visibility-Culling, Simplification, Level of Detail, Image-Based Rendering und weitere vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden können die wichtigsten Techniken im Bereich der Echtzeit-Darstellung virtueller 3D-Szenen anwenden. Sie können entscheiden, in welcher virtuellen 3D-Szene welcher Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
<p>Algorithmen für hochkomplexe virtuelle Szenen:</p> <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben</li> <li>• Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt</li> <li>• In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Mündliche Prüfung (ca. 40 Minuten)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: schriftliche Übungsaufgaben  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Matthias Fischer
<b>Lernmaterialien, Literaturangaben</b>
<p>Algorithmen für hochkomplexe virtuelle Szenen: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter</p> <ul style="list-style-type: none"> <li>• Real-Time Rendering; Tomas Akenine-Möller, Eric Haines; AK Peters, 2002.</li> <li>• Level of Detail for 3D Graphics; David Luebke, Martin Reddy, Jonathan D. Cohen; Morgan Kaufmann Publishers, 2002.</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.10 Wahlpflichtmodul: Algorithms for Synthesis and Optimization of Integrated Circuits

Modulname	Algorithms for Synthesis and Optimization of Integrated Circuits / Algorithms for Synthesis and Optimization of Integrated Circuits
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Algorithms for Synthesis and Optimization of Integrated Circuits : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithms for Synthesis and Optimization of Integrated Circuits: Vorlesung ( 45h / 105h / EN / SS / 30 )	
Algorithms for Synthesis and Optimization of Integrated Circuits: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithms for Synthesis and Optimization of Integrated Circuits: Kenntnisse aus Digitaltechnik sind hilfreich.	
<b>Inhalte</b>	
Algorithms for Synthesis and Optimization of Integrated Circuits: Die Veranstaltung behandelt die wesentlichen Schritte bei der Synthese digitaler Schaltungen und geht speziell auf die Übersetzung von Beschreibungen in Hardwarebeschreibungssprachen in Schaltungen ein. Weiterhin werden die wichtigsten Techniken für die Logikoptimierung diskutiert. In praktischen Übungen wird die effiziente Verwendung von Entwurfswerkzeugen geübt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>zwischen den verfügbaren Optimierungsmethoden für den digitalen Schaltungsentwurf auszuwählen,</li> <li>die wesentlichen Probleme bei Entwurf integrierter Schaltungen zu identifizieren und die Tradeoffs beim Schaltungsentwurf zu erkennen, und</li> <li>aktuelle Werkzeuge für den digitalen Schaltungsentwurf zu bewerten.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Gruppenarbeit</li> <li>Lernkompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
Algorithms for Synthesis and Optimization of Integrated Circuits: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Rechnerübungen mit Hardwaresynthesewerkzeugen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Hassan Ghasemzadeh Mohammadi
<b>Lernmaterialien, Literaturangaben</b>
Algorithms for Synthesis and Optimization of Integrated Circuits: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• Micheli, Giovanni De. Synthesis and optimization of digital circuits. McGraw-Hill Higher Education, 1994.</li> <li>• Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine



### 3.11 Wahlpflichtmodul: Architektur paralleler Rechnersysteme

Modulname	Architektur paralleler Rechnersysteme / Architectures of Parallel Computer Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Architektur paralleler Rechnersysteme : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Architektur paralleler Rechnersysteme: Vorlesung ( 45h / 105h / DE / SS / 20 ) Architektur paralleler Rechnersysteme: Übung ( 30h / 0h / DE / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Architektur paralleler Rechnersysteme: Grundlagen der Rechnerarchitektur	
<b>Inhalte</b>	
Architektur paralleler Rechnersysteme: Diese Veranstaltung führt in Rechnerarchitekturen der wichtigsten Parallelrechner und in die Nutzung dieser Systeme ein. Der Schwerpunkt der Veranstaltung liegt auf High-Performance-Computer.	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Studierende benennen und erklären Programmierparadigmen paralleler Programmiersprachen. Sie beherrschen grundlegende Sprachkonstrukte und Bibliotheksfunktionen wichtiger paralleler Programmiersprachen/-umgebungen (z.B. OpenMP, POSIX-Threads, MPI, PGAS) und können deren Einsatzgebiete benennen. Studierende sind in der Lage sowohl einige aktuelle HPC-Systeme als auch moderne Prozessoren mit deren Eigenschaften zu beschreiben. Sie erkennen bedeutende Trends (Power Wall, Memory Wall, ILP Wall) denen diese Systeme unterliegen. Studierende benennen und erklären allgemein genutzte Klassifikation von Parallelrechnern. Sie erklären die wichtigsten Strukturbausteine und Operationsprinzipien paralleler Rechnersysteme. Sie beherrschen die theoretische Beschreibung des Skalierungsverhaltens (Amdahl, Gustafson) und die quantitativen Bewertungen von Parallelrechnern. Studierende benennen und erklären Architekturmerkmale skalierbarer speichergekoppelte Systeme. Sie beherrschen unterschiedliche Techniken zur Aufrechterhaltung der Speicherkonsistenz und -kohärenz in busbasierten Systemen (Invalidierungs-, Update-Protokolle). Sie sind in der Lage Techniken zur Steigerung der Leistungsfähigkeit dieser Systeme zu beschreiben (Multi-Level-Caches, transiente Zustände, Split-Transaktion-Busse). Studierende erklären Mechanismen zur Synchronisation (Locks, Barrieren) in Parallelrechnern. Studierende demonstrieren Kenntnisse in Aufrechterhaltung der Cache-Kohärenz von skalierbaren Rechnersystemen (hierarchisches Snooping, Directories). Sie beherrschen</p>	

<p>Techniken zur Steigerung der Leistungsfähigkeit solcher Systeme (z.B. Latenz-Verbesserung, Durchsatzserhöhung). Studierende beschreiben Verfahren basierend auf Token Coherence. Studierende benennen und erklären grundlegende Eigenschaften von Cluster-Architekturen. Sie können die in dem Bereich eingesetzte Kommunikationsnetzwerke topologisch beschreiben und bewerten (z.B. Grad, Durchmesser, Bisektion). Sie beherrschen Kommunikationstechniken der Hochgeschwindigkeitsnetzwerke (Wormhole Routing, Virtual Cut-Through) und Routing-Verfahren (tabellenbasiertes Routing, Source-Routing). Sie beherrschen Beweistechniken zur Sicherstellung der Deadlock-Freiheit von Routings. Studierende können die Eigenschaften existenter Interconnects (z.B. InfiniBand, OmniPath) benennen. Die Studierenden sind in der Lage, wesentliche Architektureigenschaften von parallelen Rechnersystemen zu erkennen und deren Eignung für bestimmte Anwendungsgebiete festzustellen. Die Kenntnisse können dazu eingesetzt werden um hohe Rechenleistungen auf HPC-Systemen zu erzielen und vorhandene Ressourcen effizient zu nutzen.</p>
<p><b>Nichtkognitive Kompetenzen</b></p>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>
<p><b>Methodische Umsetzung</b></p>
<p>Architektur paralleler Rechnersysteme: Einsatz von Folien. In der Übung wird ein Zugang zu vorhandenen HPC-Systemen genutzt um den praktischen Umgang mit den Rechnern zu üben und die Kenntnisse der Vorlesung zu vertiefen.</p>
<p><b>Prüfungsleistung (Dauer)</b></p>
<p>Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<p><b>Moduleilprüfungen</b></p>
<p>keine</p>
<p><b>Studienleistung / qualifizierte Teilnahme</b></p>
<p>Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<p><b>Voraussetzungen für die Teilnahme an der Prüfung</b></p>
<p>Bestehen der Studienleistung.</p>
<p><b>Voraussetzungen für die Vergabe von Credits</b></p>
<p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>
<p><b>Gewichtung für die Gesamtnote</b></p>
<p>Das Modul wird mit 6 Credits gewichtet.</p>
<p><b>Modul wird in folgenden Studiengängen verwendet</b></p>
<p>–</p>

<b>Modulbeauftragte/r</b>
Dr. Jens Simon
<b>Lernmaterialien, Literaturangaben</b>
Architektur paralleler Rechnersysteme: Foliensatz
<b>Sonstige Hinweise</b>
keine

### 3.12 Wahlpflichtmodul: Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies

Modulname	Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies / Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Vorlesung ( 45h / 105h / EN / SS / 30 )	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Grundlegende Kenntnisse der Kryptographie sind empfehlenswert, jedoch keine Voraussetzung.	
<b>Inhalte</b>	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Diese Vorlesung vermittelt Techniken zur Konstruktion dezentraler digitaler Währungen wie z.B. Bitcoin, ein Verständnis für die erreichbaren und erreichten Sicherheitseigenschaften, sowie kryptographische Werkzeuge zum Schutz der Privatsphäre in solchen Anwendungen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende kennen die grundlegenden Konzepte, die modernen verteilten kryptographischen Währungen zugrunde liegen und können diese erklären und anwenden. Dies umfasst kryptographische Hashfunktionen, digitale Signaturverfahren, und Blockchains. Sie kennen die nötigen Sicherheitsanforderungen an diese Bausteine und können einschätzen, inwiefern ein gegebenes Verfahren diese Anforderungen erfüllt. Sie kennen die Begriffe Pseudonymität und Anonymität und ihre genaue Bedeutung und Abgrenzung, sind in der Lage zu untersuchen inwiefern eine gegebene Anwendung diese Eigenschaften besitzt, kennen fortgeschrittene kryptographische Techniken um starke Anonymität zu erreichen, wie zum Beispiel Ring-Signaturen, und wissen wie diese Techniken in modernen Anwendungen eingesetzt werden um die Privatheit von Nutzern zu schützen.	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr., Tibor Jager
<b>Lernmaterialien, Literaturangaben</b>
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Tafelanschrieb</li> <li>• Originalliteratur</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.13 Wahlpflichtmodul: Build It, Break It, Fix It

Modulname	Build It, Break It, Fix It / Build It, Break It, Fix It
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Build It, Break It, Fix It : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Build It, Break It, Fix It: Vorlesung ( 45h / 105h / EN / SS / 20 ) Build It, Break It, Fix It: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Bachelor in Informatik (oder gleichwertig)	
<b>Empfohlene Kenntnisse</b>	
Build It, Break It, Fix It: Fortgeschrittenenkenntnisse der Java Programmiersprache und Software-Sicherheitsanforderungen. Best practices im Bereich der sicheren Softwareentwicklung und Erfahrungen mit dem Auffinden und Ausnutzen von Sicherheitslücken in Software sind hilfreich.	
<b>Inhalte</b>	
<p>Build It, Break It, Fix It: Das Ziel dieser Lehrveranstaltung ist die praktische Vermittlung von Grundprinzipien der sicheren Softwareentwicklung. Sie ist inspiriert vom "Break It, Build It, Fix It Wettbewerb von Ruef et al. [1].</p> <p>Die Veranstaltung ist in drei Phasen aufgeteilt, in denen die Teilnehmenden in Gruppen ihre Fertigkeiten in der Entwicklung von Software, Identifizierung von Sicherheitslücken und Behebung derselben unter Beweis stellen und weiterentwickeln.</p> <p>In der "Build ItPhase entwickeln die Gruppen kleine Softwareprojekte nach einer formalen Spezifikation, die auch Sicherheitsanforderungen enthält. In der "Break ItPhase werden die entwickelten Softwareprodukte unter den Gruppen ausgetauscht mit der Zielsetzungen, Schwachstellen in anderen Implementierungen zu finden und auszunutzen. In der "Fix ItPhase behebt jede Gruppe die in ihrer Software gefundenen Schwachstellen.</p> <p>Die Veranstaltung enthält einen theoretischen Teil, in dem grundsätzliche Vorgehensweisen zur sicheren Softwareentwicklung erläutert, sowie verschiedene Arten von Sicherheitslücken vorgestellt und demonstriert werden. Der Fokus dieser Veranstaltung liegt jedoch auf der praktischen Arbeit in den Gruppen. Da das Finden und Beheben von Schwachstellen in Software eine Vielzahl von Fertigkeiten und einiges an Kreativität erfordert, sollten die Studenten ein gleiches Maß an Eigenmotivation und Selbstorganisation mitbringen.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	

Theoretische/Praktische Kenntnisse im Bereich der sichere Softwareentwicklung, Theoretische/Praktische Kenntnisse in der Auffindung und Ausnutzung von Software-Sicherheitslücken Wissen über verbreitete, reale Software-Sicherheitslücken und Möglichkeiten diese auszunutzen.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Build It, Break It, Fix It:
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: Praktikumsarbeit Qualifizierte Teilnahme: Praktikumsarbeit mit anschließendem Gespräch Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eric Bodden
<b>Lernmaterialien, Literaturangaben</b>
Build It, Break It, Fix It: Das Kursmaterial wird auf der KoaLA Seite des Kurses angeboten werden.



<b>Sonstige Hinweise</b>
keine

### 3.14 Wahlpflichtmodul: Clustering Algorithms

Modulname	Clustering Algorithms / Clustering Algorithms
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Clustering Algorithmen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Clustering Algorithmen: Vorlesung ( 45h / 105h / EN / WS oder SS / 25 ) Clustering Algorithmen: Übung ( 30h / 0h / EN / WS oder SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Clustering Algorithmen: Algorithmen und Datenstrukturen, Lineare Algebra, Wahrscheinlichkeitstheorie	
<b>Inhalte</b>	
Clustering Algorithmen: Im Zentrum dieser Vorlesung steht eines der wichtigsten Werkzeuge der Datenanalyse. das Clustering. Clustering ist der Prozess des Aufteilen von Daten in sinnvolle oder nützliche Teilmengen. Eine solche Aufteilung sollte die natürliche Struktur der Daten widerspiegeln. Dies kann z.B. bedeuten, dass jede Teilmenge möglichst viele einander ähnliche Objekte einer Datenmenge enthalten soll. Clustering ist eine natürliche Art Daten zu strukturieren und zu analysiere. Es besitzt viele Anwendung in den Natur- und Ingenieurwissenschaften.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende können die wichtigsten Algorithmentechniken des Clustering benennen und erläutern. Sie können entscheiden, in welcher Situation welcher Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen. Sie kennen verschiedene Modellierungen von Clustering, können diese anwenden und anpassen sowie für die Modellierung geeignete Algorithmen einsetzen und bewerten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Clustering Algorithmen: Vorlesung mit Übungen, Lesegruppen, Kurzvorträge	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Clustering Algorithmen: David J.C MacKay, Information Theory, Inference and Learning Algorithms Christopher M. Bishop, Pattern Recognition and Machine Learning Folien der Vorlesung
<b>Sonstige Hinweise</b>
keine

### 3.15 Wahlpflichtmodul: Compiler Construction

Modulname	Compiler Construction / Compiler Construction
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Compilerbau : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Compilerbau: Vorlesung ( 45h / 105h / EN / WS / 25 ) Compilerbau: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Compilerbau: Elementare Kenntnisse aus dem Compilerbau, wie etwa aus der LVA "Programming Languages and Compilers" (PLaC)	
<b>Inhalte</b>	
Compilerbau: Die Vorlesung setzt da an, wo "Programming Languages and Compilers" (PLaC, Bachelor Studium) aufgehoert hat. Thematisch steht somit das "Compiler Backend" im Vordergrund und es werden theoretische Grundlagen wichtiger Optimierungen im Compilerbau vermittelt. Die theoretischen Grundlagen werden dann durch einen Praxis-Teil veranschaulicht, bei dem konkrete Implementierungen im LLVM Compiler Framework diskutiert werden.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studenten erwerben fundierte Kenntnisse zur Theorie und Praxis des Compilerbaus. Der vermittelten Grundlagen aus der Programm-Analyse sowie darauf folgende Transformationen/Optimierungen erlauben es dem Studenten, das Optimierungspotential ihrer eigenen Programme voll auszuschöpfen. Der praktische Teil veranschaulicht konkrete Implementierungsdetails und vertieft daher die theoretischen Grundlagen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Compilerbau: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb.</li> <li>• Präsenzübungen in kleinen Gruppen</li> </ul>	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof.Dr. Stefan Brunthaler
<b>Lernmaterialien, Literaturangaben</b>
Compilerbau: <ul style="list-style-type: none"> <li>• Appel: Modern Compiler Construction in ML.</li> <li>• Cooper, Torczon: Engineering a Compiler.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.16 Wahlpflichtmodul: Computational Geometry

Modulname	Computational Geometry / Computational Geometry
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Algorithmische Geometrie : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithmische Geometrie: Vorlesung ( 45h / 105h / EN / SS / 40 ) Algorithmische Geometrie: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithmische Geometrie: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen wird angenommen.	
<b>Inhalte</b>	
Algorithmische Geometrie: Es werden Algorithmen und Datenstrukturen aus dem Bereich der Algorithmischen Geometrie behandelt. Die Grundelemente und Eingabe sind geometrische Daten (Punkte, Linien, Kreise, Polygone, Körper). Die Probleme werden geometrisch formuliert und dafür wird eine algorithmische Lösung mit Hilfe spezieller geometrische Datenstrukturen gesucht. Die Algorithmen werden theoretisch analysiert. Dazu wird Laufzeit und Speicherplatz bestimmt und die Korrektheit der Algorithmen bewiesen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden können die wichtigsten Techniken im Bereich der Algorithmischen Geometrie anwenden. Sie können entscheiden, für welche geometrisch formulierten Probleme welcher Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Algorithmische Geometrie: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> </ul>	

<ul style="list-style-type: none"> <li>• erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben</li> <li>• Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt</li> <li>• In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Matthias Fischer
<b>Lernmaterialien, Literaturangaben</b>
Algorithmische Geometrie: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter <ul style="list-style-type: none"> <li>• Computational Geometry: Algorithms and Applications Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars, Springer-Verlag, 2008</li> <li>• Algorithmische Geometrie Rolf Klein, Springer-Verlag, 2005</li> <li>• Lectures on Discrete Geomtetry Jiri Matousek, Springer-Verlag, 2002.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.17 Wahlpflichtmodul: Deductive Verification

Modulname	Deductive Verification / Deductive Verification
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Deductive Verification : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Deductive Verification: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Deductive Verification: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Deductive Verification: Aussagen- und Prädikatenlogik. Imperative Programmierung.	
<b>Inhalte</b>	
Deductive Verification: In der Vorlesung werden Techniken zum Nachweis der Korrektheit von Programmen vorgestellt. Im Gegensatz zu der Model Checking Vorlesung basieren diese Techniken auf Beweiskalkülen, nicht auf automatischen Methoden. Die Vorlesung wird zuerst das Konzept der Zusicherungen einführen und am Beispiel von Java Programmen illustrieren. Danach werden Hoare-Kalküle zum Nachweis von Zusicherungen erläutert, und zwar sowohl für sequentielle als auch parallele Programme. Für die Beweiskalküle wird ihre Korrektheit und Vollständigkeit untersucht. Zum Schluss werden Beweiskalküle für temporal-logische Formeln vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden kennen die wesentlichen Prinzipien der Programmverifikation. Sie können für sequentielle wie parallele Programme nachweisen, dass diese korrekt sind. Sie wissen wie Zusicherungen zur Beschreibung der Funktionalität von Programmen genutzt werden können. Sie kennen eine temporale Logik und können diese zur Spezifikation von Eigenschaften einsetzen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	



Deductive Verification: Eine Mischung aus Folien und Tafelanschrieb. Interaktion mit Studierenden über Fragen und Beispielaufgaben. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Deductive Verification: K. Apt, E.-R. Olderog: Verification of sequential and concurrent programs Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.18 Wahlpflichtmodul: Designing code analyses for large-scale software systems

Modulname	Designing code analyses for large-scale software systems / Designing code analyses for large-scale software systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Designing code analyses for large-scale software systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Designing code analyses for large-scale software systems: Vorlesung ( 45h / 105h / EN / WS / 30 ) Designing code analyses for large-scale software systems: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Designing code analyses for large-scale software systems: Der vorherige Besuch der Veranstaltung Software Analysis wird empfohlen ist aber nicht zwingend erforderlich. Ein gutes Verständnis von Java und den Prinzipien objektorientierter Programmierung ist hilfreich.	
<b>Inhalte</b>	
Designing code analyses for large-scale software systems: Statische Codeanalysen dienen dazu, automatisiert Fehler und Schwachstellen im Programmcode aufzufinden. Zu diesem Zwecke suchen sie nach bekannten Fehlermustern. In dieser Vorlesung wird erklärt, wie man solche Codeanalysen entwirft, die inter-prozedural sind, also das komplette Programm betrachten, über die Grenzen einzelner Prozeduren hinweg. Der Entwurf solcher Analysen gestaltet sich deshalb sehr schwierig, weil die Analysen oft Millionen von Programmstatements gleichermaßen präzise aber auch effizient verarbeiten müssen. Es werden außerdem Beispielsanalysen aus dem Bereich der IT-Sicherheit besprochen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Durch den Besuch erlernen Studierende... - wichtige Designentscheidungen beim Entwurf automatisierter Codeanalysen richtig zu treffen - welche Algorithmen für Codeanalysen in welchen Anwendungssituationen am besten geeignet sind - wie man Codeanalysen für reale Probleme aus der IT-Sicherheit entwirft - wie man gängige Begriffliche wie Kontext-, Fluss-, Feld-, und Objekt-Sensitivität korrekt interpretiert - welche Limitierungen statische Codeanalysen aufweisen - welche gängige Codeanalysen für Sicherheitsschwachstellen (OWASP Top 10 etc.) existieren, und wie sich diese mit den vorgestellten Algorithmen umsetzen lassen.	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Designing code analyses for large-scale software systems: Vorlesung und Gruppenübungen
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eric Bodden
<b>Lernmaterialien, Literaturangaben</b>
Designing code analyses for large-scale software systems: Empfohlene Literatur: * Thomas Reps, Susan Horwitz, and Mooly Sagiv. 1995. Precise interprocedural dataflow analysis via graph reachability. POPL '95 * Shmuel Sagiv, Thomas W. Reps, and Susan Horwitz. 1995. Precise Interprocedural Dataflow Analysis with Applications to Constant Propagation. TAPSOFT '95 * Akash Lal, Thomas Reps, and Gogul Balakrishnan. 2005. Extended weighted pushdown systems. CAV 2005 * Nomair A. Naeem, Ondrej Lhoták, and Jonathan Rodriguez. 2010. Practical extensions to the IFDS algorithm. CC 2010 * Yannis Smaragdakis, Martin Bravenboer, and Ondrej Lhoták. 2011. Pick your contexts well: understanding object-sensitivity. POPL 2011 * Eric Bodden. 2012. Inter-procedural data-flow analysis with IFDS/IDE and Soot. SOAP 2012 * Rohan Padhye, Uday P. Khedker. Interprocedural Data Flow Analysis in Soot using Value Contexts. SOAP 2013

<b>Sonstige Hinweise</b>
keine

### 3.19 Wahlpflichtmodul: Empiric performance evaluation

Modulname	Empiric performance evaluation / Empiric performance evaluation
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Empirische Leistungsbewertung : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Empirische Leistungsbewertung: Vorlesung ( 45h / 105h / EN / SS / 15 ) Empirische Leistungsbewertung: Übung ( 30h / 0h / EN / SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Empirische Leistungsbewertung: Stochastik auf dem Niveau der Bachelor-Ausbildung.	
<b>Inhalte</b>	
Empirische Leistungsbewertung: Die Vorlesung beschreibt Methoden und Verfahren, um experimentelle und simulationsbasierte Leistungsbewertung durchzuführen und statistisch korrekt auszuwerten. Die erlernten Verfahren sind auf eine weite Klasse von Systemen anwendbar.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer können bestimmen, ob ein gegebenes System/Modell einer bestimmten Leistungsbewertungsmethode zugänglich ist. Sie können ein Experiment oder eine Simulation entwerfen und durchführen, die geeigneten stochastischen Modelle auswählen und die Ergebnisse korrekt interpretieren. Sie können statistisch gerechtfertigte Schlüsse ziehen, z.B. ob eines von mehreren System als das beste System angesehen werden kann.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Empirische Leistungsbewertung: Vorlesung mit Folien und Tafelanschrieb; Übungsblätter.	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Empirische Leistungsbewertung: Foliensatz, Übungsblätter, Lehrbuch Kelton & Law, Simulation Modelling and Analysis.
<b>Sonstige Hinweise</b>
keine

### 3.20 Wahlpflichtmodul: Foundations of Cryptography

Modulname	Foundations of Cryptography / Foundations of Cryptography
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Foundations of Cryptography : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Foundations of Cryptography: Vorlesung ( 45h / 105h / EN / SS / 25 ) Foundations of Cryptography: Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Foundations of Cryptography: Basiskenntnisse in IT-Sicherheit und Kryptographie nützlich aber nicht notwendig, Grundkonzepte der Komplexitätstheorie und Wahrscheinlichkeitstheorie	
<b>Inhalte</b>	
Foundations of Cryptography: Wichtige Basiskonzepte moderner Kryptographie werden vorgestellt. Hierzu gehören Verschlüsselungsverfahren, digitale Signaturen, Identifikationsprotokolle und Mehrparteienberechnungen werden vorgestellt. In allen Fällen werden formale Sicherheitsdefinitionen vorgestellt und, ausgehend von mathematisch präzisen Annahmen, beweisbar sichere Konstruktionen entwickelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende verstehen wesentliche Konzepte und Methoden moderner Kryptographie. Sie können für für Sicherheitsprobleme geeignete kryptographische Techniken auswählen. Sie können Basistechniken der Kryptographie kombinieren und modifizieren, neue Sicherheitskonzepte definieren und die Sicherheit der Konstruktionen bezüglich dieses Definitionen beweisen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Foundations of Cryptography: Vorlesung mit Übungen, Lesegruppen	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Foundations of Cryptography: Oded Goldreich, Foundations of Cryptography I,II, Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography Folien der Vorlesung
<b>Sonstige Hinweise</b>
keine



### 3.21 Wahlpflichtmodul: Fundamentals of Model-Driven Engineering

Modulname	Fundamentals of Model-Driven Engineering / Fundamentals of Model-Driven Engineering
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Fundamentals of Model-Driven Engineering : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Fundamentals of Model-Driven Engineering: Vorlesung ( 45h / 105h / EN / WS / 30 ) Fundamentals of Model-Driven Engineering: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Fundamentals of Model-Driven Engineering:	
<b>Inhalte</b>	
<p>Fundamentals of Model-Driven Engineering: Die Verwendung von Modellen als primäre Artefakte im Softwareentwicklungsprozess ist eines der Hauptziele der modellgetriebenen Softwareentwicklung (Englisch: Model-Driven Engineering (MDE)). Um dieses Ziel zu erreichen müssen einige Aufgaben gut unterstützt werden: Die Spezifikation neuer Modellierungssprachen (Metamodellierung), die Erzeugung sowie Manipulation von Modellen (Modelltransformationen), und die Möglichkeit Änderungen an einem Modell auf andere betroffene Modellen zu propagieren (Modellsynchronisierung).</p> <p>Nicht nur ein intuitives Verständnis für diese zentralen Konzepte ist wichtig, sondern auch eine präzise Formalisierung wenigstens der wichtigsten Grundkonzepte. Dies ist vor allem dann unabdingbar, wenn man verlässliche Werkzeugunterstützung bieten möchte. In der Vorlesung werden daher Grundkonzepte des MDE eingeführt und mit einfachen Mitteln der Kategorientheorie formalisiert.</p> <p>Die Vorlesung ist mit dem Ziel entworfen besonders Informatikern zugänglich zu sein. Dies wird erreicht durch eine konstruktive und direkte Abbildung aller Definitionen und Beweiskonstruktionen auf lauffähige Programme in Java.</p> <p>Die Vorlesung ergänzt die MDS-D-Vorlesung so, dass Studierende beide Vorlesungen in beliebiger Reihenfolge besuchen können.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende sind in der Lage, selbständig weiterführende Bücher und Papiere über Graphtransformationen zu lesen und neue Definitionen sowie Konstruktionsverfahren zu verstehen. Studierende können die Methoden zur Formalisierung aus der Vorlesung anwenden um neue Strukturen und die regelbasierte Ma-	

nipulation dieser Strukturen analog zu formalisieren. Studierende verstehen und schätzen die Vorteile einer präzisen und konstruktiven Formalisierung von Pattern-Matching, Constraints, und Regeln im Kontext der modellgetriebenen Softwareentwicklung. Studierende kennen einige Grundkonzepte der Kategorientheorie und verstehen wie und wieso diese für die Formalisierung von Graphtransformationen verwendet werden.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Fundamentals of Model-Driven Engineering: Mischform einer Frontalveranstaltung (kurze Vorlesungseinheiten) und Übung. Übungsaufgaben werden während der Veranstaltung in Gruppen bearbeitet und diskutiert. Jede Übung besteht aus einem theoretischen und einem praktischen Teil. Der praktische Teil baut auf einem in Java programmierten Framework auf, das für die Vorlesung konzipiert und implementiert wurde.</p> <p>Die letzten 4-5 Termine der Veranstaltung werden für eine themenspezifische Vertiefung der Grundinhalte verwendet. Es werden Gruppen von 3-4 Studierenden gebildet, weiterführende Themen besprochen, gewählt, und als "Mini-Praktikum" parallel bearbeitet. Ergebnisse werden bei dem letzten Termin allen anderen Gruppen präsentiert. In dieser letzten Phase der Veranstaltung sollen Studierende vor allem die selbständige Einarbeitung und Erarbeitung von neuem weiterführendem Wissen üben.</p>
<b>Prüfungsleistung (Dauer)</b>
<p>Mündliche Prüfung (ca. 40 Minuten)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: schriftliche Übungsaufgaben  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.

<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Jun.-Prof.Dr. Anthony Anjorin
<b>Lernmaterialien, Literaturangaben</b>
Fundamentals of Model-Driven Engineering: Ehrig, H., Ehrig, K., Prange, U., & Taentzer, G. (2006). Fundamentals of Algebraic Graph Transformation. (W. Brauer, G. Rozenberg, & A. Salomaa, Eds.). Springer. Awodey, S. (2006). Category Theory. Ebsco Publishing.
<b>Sonstige Hinweise</b>
keine

### 3.22 Wahlpflichtmodul: Future Internet

Modulname	Future Internet / Future Internet
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Future Internet : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Future Internet: Vorlesung ( 45h / 105h / EN / SS / 20 ) Future Internet: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Future Internet: Grundkenntnisse in Rechnernetze (z.B. durch die Bachelor-Vorlesung Rechnernetze).	
<b>Inhalte</b>	
Future Internet: Die Veranstaltung diskutiert aktuelle, forschungsnahe Entwicklung des Internets und der Vernetzung von Rechenzentren. Sie wird dynamisch an entsprechende Themen angepasst und basiert insbesondere auf wissenschaftlichen Veröffentlichungen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer werden durch die Vorlesung an den aktuellen Stand der Internet-Forschung herangeführt. Sie kennen die Schwachpunkte der aktuellen Architektur, können diese geeignet kritisieren und können diese mit aktuellen Vorschlägen kontrastieren sowie Vor- und Nachteile der jeweiligen Lösungen bewerten. Sie können für unterschiedliche Nutzungssituationen die Anwendbarkeit einer bestimmten Lösung einschätzen und voraussagen. Methodisch sind sie in der Lage, Netz-Experimente zu entwerfen und durchzuführen. Teilnehmer können neue Vorschläge für Architekturen und Protokolle kreieren, diese mit anderen Ansätzen vergleichen und bewerten, und sich für eine geeignete Lösung entscheiden. Da die Vorlesung auf aktuellen Veröffentlichungen beruht, sind Teilnehmer in der Lage, sich selbständig in neues, nicht didaktisch aufbereitetes Material einzuarbeiten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	

Future Internet: Vorlesung mit Folien und Tafelanschrieb; begleitende Übungen. Teilnehmer werden zu eigener Literaturstudie aktueller Veröffentlichungen angehalten. In der Übung werden Architekturexperimente, bspw. mit OpenFlow, durchgeführt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Future Internet: Foliensatz, insbesondere auch aktuelle Veröffentlichungen. Kein umfassendes Lehrbuch verfügbar, Teile abgedeckt durch Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud.
<b>Sonstige Hinweise</b>
keine

### 3.23 Wahlpflichtmodul: Hardware/Software Codesign

Modulname	Hardware/Software Codesign / Hardware/Software Codesign
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Hardware/Software Codesign : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Hardware/Software Codesign: Vorlesung ( 45h / 105h / EN / SS / 40 ) Hardware/Software Codesign: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Hardware/Software Codesign: Digitaltechnik, Rechnerarchitektur, Programmierkenntnisse	
<b>Inhalte</b>	
Hardware/Software Codesign: Hardware/Software Codesign bezeichnet den integrierten und automatisierten Entwurf von Hard- und Software in Computersystemen. Die Lehrveranstaltung Hardware/Software Codesign lehrt Konzepte und Methoden, welche in computerunterstützten Entwurfswerkzeugen zur Entwurfsraumexploration, Entwurfsoptimierung und Compilation für spezialisierte Computersysteme verwendet werden.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer dieser Lehrveranstaltung können die Ziele und Herausforderungen beim Entwurf von spezialisierten Computersystemen benennen. Sie können die passenden Modellierungsansätze für ein gegebenes HW/SW System und entsprechenden funktionalen und nichtfunktionalen Anforderungen auswählen und eine Spezifikation im gewählten Formalismus erstellen. Sie können Zielarchitekturen für die Implementierung von HW/SW Systemen charakterisieren und die Eignung einer spezifischen Zielarchitektur für eine gegebene Anwendung bewerten. Sie können den Aufbau eines Compilers beschreiben, Grundblöcke und Kontrollflussgraphen verstehen und anwenden, sowie Optimierungs- und Codegenerierungsmethoden diskutieren. Sie können demonstrieren, wie Programme mittels High-level Synthesemethoden in Hardware übersetzt werden. Sie verstehen die Methode der ganzzahlig linearen Programmierung und können sie auf Probleme aus den Bereichen Synthese, Ablaufplanung und Software Performanceschätzung anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	

<b>Methodische Umsetzung</b>
Hardware/Software Codesign: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Selbststudium und Diskussion wissenschaftlichen Publikationen</li> <li>• Praktische Programmierprojekte zur Umsetzung und Anwendung von Methoden aus dem Bereich Hardware/Software Codesign</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Christian Plessl
<b>Lernmaterialien, Literaturangaben</b>
Hardware/Software Codesign: Folien, Standardlehrbücher, Übungsblätter, Wissenschaftliche Artikel
<b>Sonstige Hinweise</b>
keine

### 3.24 Wahlpflichtmodul: High-Performance Computing

Modulname	High-Performance Computing / High-Performance Computing
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• High-Performance Computing : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
High-Performance Computing: Vorlesung ( 30h / 105h / EN / WS / 40 ) High-Performance Computing: Übung ( 45h / 0h / EN / WS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
High-Performance Computing: <ul style="list-style-type: none"> <li>• Programmierkenntnisse in C/C++</li> <li>• Rechnerarchitektur</li> </ul>	
<b>Inhalte</b>	
High-Performance Computing: Die Veranstaltung vermittelt Grundlagen des Hochleistungsrechnen (High-Performance Computing) mit einem Schwerpunkt auf der Programmierung von parallelen Rechensystemen und neuartiger Hardwarebeschleuniger.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• Modelle und Programmiermuster für HPC zu benennen und die passenden Muster für eine gegebenen Anwendung zu identifizieren,</li> <li>• die Grundkonstrukte der gängigen HPC Bibliotheken, insbesondere MPI, OpenMP und OpenCL, anzugeben und anzuwenden,</li> <li>• die Performance von Anwendungen durch Verwendung von Profilingwerkzeugen zu analysieren und systematisch passende Optimierungsstrategien abzuleiten,</li> <li>• die gelernten Konzepte und Verfahren auf existierende Anwendungen anzuwenden und diese zu parallelisieren und optimieren.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> </ul>	
<b>Methodische Umsetzung</b>	
High-Performance Computing:	



<ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Praktische Programmierprojekte auf Parallelrechnersystemen in Kleingruppen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Christian Plessl
<b>Lernmaterialien, Literaturangaben</b>
High-Performance Computing: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Programmierprojekte</li> <li>• Lehrbuch: Pacheco: An Introduction to Parallel Programming. Morgan Kaufmann, 2011.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.25 Wahlpflichtmodul: Intelligence in Embedded Systems

Modulname	Intelligence in Embedded Systems / Intelligence in Embedded Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Intelligenz in eingebetteten Systemen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Intelligenz in eingebetteten Systemen: Vorlesung ( 45h / 105h / EN / SS / 25 ) Intelligenz in eingebetteten Systemen: Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Intelligenz in eingebetteten Systemen:	
<b>Inhalte</b>	
<p>Intelligenz in eingebetteten Systemen: Intelligente eingebettete Systeme sind technische Systeme, die mittels unterschiedlicher Sensoren und Aktoren ihre Umwelt wahrnehmen sowie (teil-)autonom mit ihr interagieren. Häufig werden Methoden und Algorithmen der Künstlichen Intelligenz zur Steuerung des Verhaltens eingesetzt. Diese ermöglichen es den Systemen beispielsweise, ihr Verhalten zielgerichtet zu planen sowie durch Adaption und Lernen selbständig zu optimieren. Intelligente eingebettete Systeme gewinnen kontinuierlich an Bedeutung, nicht nur auf wissenschaftlicher Ebene innerhalb der Informatik, sondern auch im sozialen und gesellschaftlichen Kontext: Autonome oder teilautonome Systeme wie Serviceroboter, selbstfahrende PKWs oder medizinische Hilfs- und Diagnosesysteme werden unser privates und berufliches Leben in absehbarer Zukunft tiefgreifend verändern.</p> <p>Diese Vorlesung greift wichtige Aspekte des Entwurfs intelligenter eingebetteter Systeme auf und vermittelt entsprechende theoretische und methodische Grundlagen. Ausgehend von typischen Architekturen solcher Systeme erstrecken sich die Inhalte über Themen der intelligenten Sensorverarbeitung und Modellierung der Umwelt bis hin zur intelligenten Handlungssteuerung und Selbstadaption.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende benennen und erklären Methoden und Algorithmen zur intelligenten Sensorverarbeitung und Handlungssteuerung (z. B. Bildverarbeitung, Sensorfusion, Kartendarstellung, Navigation, Planung und maschinellem Lernen). Sie verstehen und lösen Probleme bei der Umsetzung in eingebetteten Systemen, die über eingeschränkte Ressourcen verfügen. Ferner sind sie in der Lage sich in neue Verfahren einzuarbeiten, sie zu beurteilen und sie einzusetzen, insbesondere im Kontext eingebetteter Systeme.	
<b>Nichtkognitive Kompetenzen</b>	

<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Intelligenz in eingebetteten Systemen: <ul style="list-style-type: none"> <li>• Vorlesung mit Folien</li> <li>• Interaktive Übungen, in denen die Studenten das Verständnis des Stoffes vertiefen und das Gelernte anwenden</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Bernd Kleinjohann
<b>Lernmaterialien, Literaturangaben</b>
Intelligenz in eingebetteten Systemen: Folien, Publikationen, Bücher: <ul style="list-style-type: none"> <li>• St. Russel, P. Norvig: Artificial Intelligence: A Modern Approach</li> <li>• R. Arkin: Behavior-Based Robotics</li> </ul> Weitere Literatur (Bücher, Publikationen) werden in der Vorlesung bekanntgegeben.
<b>Sonstige Hinweise</b>

### 3.26 Wahlpflichtmodul: Interactive Data Visualization

Modulname	Interactive Data Visualization / Interactive Data Visualization
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Interaktive Datenvisualisierung : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Interaktive Datenvisualisierung: Vorlesung ( 30h / 105h / EN / SS / 30 ) Interaktive Datenvisualisierung: Übung ( 15h / 0h / EN / SS / 30 ) Interaktive Datenvisualisierung: Visualisierungsprojekt ( 30h / 0 h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Interaktive Datenvisualisierung: Sicherer Umgang mit zumindest einem Grafik API.	
<b>Inhalte</b>	
<p>Interaktive Datenvisualisierung: Die Datenvisualisierung stellt Methoden und Techniken bereit, um die den Datensätzen zugrunde liegenden korrelierenden Strukturen und Beziehungen zu erfassen und zu präsentieren. Datensätze können dabei aus unterschiedlichsten Anwendungen kommen. Das Hauptziel der Präsentation ist die Kommunikation der im Datensatz enthaltenen Information in ausdrucksvoller und wirksamer Weise für den menschlichen Betrachter. In diesem Kontext ist die Visualisierung (=das Ergebnis des Visualisierungsprozesses) ein computergeneriertes Bild oder eine Folge von Bildern, wobei der Input in diesen Prozess Daten sind und das Ziel dieses Prozesses die menschliche Wahrnehmung ist. Der Visualisierungsprozess kann also auch als Abbildung von Daten in ausdrucksstarke und wirksame (visuelle) Repräsentationen für den Menschen gesehen werden.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Der Visualisierungsprozess kann von Studierenden anhand des Vier-Komponentenmodells (Reality, Data, Picture(s), User) erklärt werden. Studierende besitzen die Fähigkeit, selektive Algorithmen zur Datenanalyse und Datenvorverarbeitung (z.B. Transferfunktionen, Filterung, statistische Analysen, Sampling, Skalierung) in Rechenschritten nachzuvollziehen und mit einem modernen API umzusetzen. APIs und Tools zur Visualisierung von unterschiedlichen Datensätzen und Visualisierungszielen können bewertet werden. Für unterschiedliche Datenmodelle können mehrere mögliche Visualisierungstechniken und Interaktionstechniken benannt und erklärt werden. Studierende besitzen die Fähigkeit, ihre eigenen sowie fremde Visualisierungen auf ihre Ausdrucksfähigkeit, Wirksamkeit, Angemessenheit und Skalierbarkeit zu bewerten. Bewertungsmethoden (Evaluationsmethoden) für den Einsatz in der Entwicklung von Visualisierungssoftware können benannt und erklärt werden. Über den Design Prozess in der Visual-</p>	

<p>isierung (Abbildung von Daten-Parametern auf visuelle Variablen) kann qualifiziert diskutiert werden. Für den Visualisierungsprozess relevante Nutzer-, Daten-, und Hardwareeigenschaften sind bekannt und können in den Design Prozess eingebracht werden. Studierende demonstrieren in einem Visualisierungsprojekt ihre Fähigkeit, sich in</p> <ul style="list-style-type: none"> <li>• Anwendungskontexte (z.B. Medizin, Astronomie, Computer-Netzwerk-Analyse, Textkorpora-Analyse) schnell einzuarbeiten,</li> <li>• alleine oder im Team Lösungswege zu recherchieren, zu verstehen, und einen oder mehrere Lösungswege zu implementieren,</li> <li>• Lösungswege und resultierende Visualisierungen technisch verständlich und motivierend zu präsentieren.</li> </ul>
<p><b>Nichtkognitive Kompetenzen</b></p>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Medienkompetenz</li> </ul>
<p><b>Methodische Umsetzung</b></p>
<p>Interaktive Datenvisualisierung: Die Vorlesung nutzt Beamer und Tafel. Die Studierenden bearbeiten kurze In-Class Aufgaben und diskutieren dann mit der Dozentin über unterschiedliche Lösungen bzw. Probleme bei den Lösungen. Hausaufgaben werden nach kurzer Vorbereitung in der Zentralübung von den Studierenden alleine gelöst, ihre Lösungen werden in der Zentralübung vorgestellt und diskutiert. Im letzten Drittel der Veranstaltung wird an einem Gruppenprojekt gearbeitet. Die Studierenden präsentieren am Ende der Veranstaltung ihre Lösungen und bewerten die Projekte ihrer Kommilitonen.</p>
<p><b>Prüfungsleistung (Dauer)</b></p>
<p>Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<p><b>Moduleilprüfungen</b></p>
<p>keine</p>
<p><b>Studienleistung / qualifizierte Teilnahme</b></p>
<p>Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<p><b>Voraussetzungen für die Teilnahme an der Prüfung</b></p>
<p>Bestehen der Studienleistung.</p>
<p><b>Voraussetzungen für die Vergabe von Credits</b></p>
<p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>
<p><b>Gewichtung für die Gesamtnote</b></p>
<p>Das Modul wird mit 6 Credits gewichtet.</p>

<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gitta Domik-Kienegger
<b>Lernmaterialien, Literaturangaben</b>
Interaktive Datenvisualisierung: Foliensatz der Vorlesung; Textbuch "Interactive Data Visualization", M. Ward, G. Grinstein, D. Keim, A K Peters.
<b>Sonstige Hinweise</b>
keine

### 3.27 Wahlpflichtmodul: Kontextuelle Informatik

Modulname	Kontextuelle Informatik / Contextual Informatics
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Kontextuelle Informatik : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Kontextuelle Informatik: Vorlesung ( 30h / 105h / DE / WS / 40 ) Kontextuelle Informatik: Übung ( 45h / 0h / DE / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Kontextuelle Informatik: Kenntnisse aus dem Bachelorstudiengang Informatik oder eines vergleichbaren Studiengangs	
<b>Inhalte</b>	
<p>Kontextuelle Informatik: Informatiker entwickeln auf Zeichen basierende Produkte (Programme, Spezifikationen, Dokumentationen etc.), die einen spezifischen Gegenstandsbereich modellieren. Bei der Entwicklung solcher Produkte stellen sich vielfältige Fragen: Wie können die zu verarbeitenden Daten sowie die umzusetzenden Prozesse angemessen modelliert werden? Welche Konsequenzen ergeben sich aus der Möglichkeit, Systeme interaktiv zu gestalten? Welche Rolle wird beim Einsatz der Software den Benutzern, welche der Software zuteil? Welche Rahmenbedingungen des Einsatzkontexts sind dabei zu beachten?</p> <p>Die Veranstaltung »Kontextuelle Informatik« erörtert die im weiteren Verlauf relevanten Grundbegriffe der Informatik mit besonderem Augenmerk auf die Unterscheidung zwischen technischen Konzepten und der Nutzungssphäre. Vor diesem Hintergrund werden Theorien interaktiver Systeme betrachtet, um insbesondere die Rolle digitaler Medien für geistige Prozesse untersuchen. Bei der Entwicklung von Informatiksystemen müssen die relevanten Daten und Prozesse bis zu einem gewissen Grad antizipiert und als formales System beschrieben werden. Dies wirft Fragen auf, unter welchen Bedingungen eine solche formale Beschreibung adäquat erfolgen kann und welche Konsequenzen sich daraus in Bezug auf die Zuverlässigkeit und den verantwortbaren Einsatz von Informatiksystemen ergeben.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden lernen, theoriegeleitet die Bedeutung von interaktiven Systemen zu untersuchen. Sie verstehen, technische und nicht-technische Problemstellungen zu differenzieren und adäquat aufeinander zu beziehen. Des Weiteren werden sie in die Lage versetzt, aktuelle technologische Entwicklungen zu	

bewerten und zu vergleichen sowie Innovationspotenziale im Bereich digitaler Medien abschätzen zu können.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Medienkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>
<b>Methodische Umsetzung</b>
Kontextuelle Informatik: Vorlesungsinhalte werden im Vortragsstil vermittelt, wobei die Studierenden durch Fragen und kurze Diskussionen interagieren. In den Übungen werden die Inhalte auf der Basis der Lektüre wissenschaftlicher Literatur sowie mit Kurzreferaten durch die Studierenden vertieft.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Harald Selke
<b>Lernmaterialien, Literaturangaben</b>
Kontextuelle Informatik: Vorlesungsfolien Wardrip-Fruin, N.; Montfort, N. (eds.): The New Media Reader. Cambridge, Ma.: MIT Press, 2003. Quinn, M. J.: Ethics for the Information Age. Boston, Ma.: Pearson, 7th edition, 2016. Begleitende wissenschaftliche Literatur wird in der Vorlesung vorgestellt.



<b>Sonstige Hinweise</b>
keine

### 3.28 Wahlpflichtmodul: Language-Based Security

Modulname	Language-Based Security / Language-Based Security
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Sprachbasierte Sicherheit : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Sprachbasierte Sicherheit: Vorlesung ( 45h / 105h / EN / WS / 25 ) Sprachbasierte Sicherheit: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Sprachbasierte Sicherheit: Grundlegende Kenntnisse aus den Bereichen Compilerbau, Programmiersprachen und IT Sicherheit. (Aus den jeweiligen Bachelor Vorlesungen “Programming Languages and Compilers” und “IT Sicherheit”.)	
<b>Inhalte</b>	
Sprachbasierte Sicherheit: Ziel des Kurses ist es, Grundlagen aus der sprach-basierten Sicherheit aus praktischer und theoretischer Sicht zu vermitteln.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studenten erwerben fundierte Kenntnisse zu aktuellen Angriffen und Verteidigungstechniken. Behandelte Techniken werden sowohl theoretisch als auch praktisch behandelt, sodass Studenten neben Faktenwissen zu den jeweiligen Techniken auch jene Methodenkompetenzen erwerben, die es ihnen erlaubt Sicherheitsfragestellungen aus Programmiersprachen-Sicht kompetent zu beantworten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Sprachbasierte Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb.</li> <li>• Begleitende Implementierung als Übungsteil.</li> </ul>	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof.Dr. Stefan Brunthaler
<b>Lernmaterialien, Literaturangaben</b>
Sprachbasierte Sicherheit: keine
<b>Sonstige Hinweise</b>
keine

### 3.29 Wahlpflichtmodul: Linear and Integer Optimization

Modulname	Linear and Integer Optimization / Linear and Integer Optimization
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Linear and Integer Optimization : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Linear and Integer Optimization: Vorlesung ( 45h / 105h / EN / WS / 30 ) Linear and Integer Optimization: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Linear and Integer Optimization: Algorithmen und Datenstrukturen Lineare Algebra	
<b>Inhalte</b>	
<p>Linear and Integer Optimization: Optimierungsprobleme sind heute allgegenwärtiger Bestandteil des täglichen Lebens in Industrie, Handwerk und Handel. Dabei geht es zumeist um die optimale Nutzung teurer Ressourcen, z.B. bei Transport-, Scheduling-, Planungs- oder Designproblemen. Viele dieser Optimierungsprobleme sind NP-hart und daher nicht effizient lösbar. Überraschenderweise lassen sich viele dieser Optimierungsprobleme anschaulich als lineare Programme (LP) oder ganzzahlig-lineare Programme (IP) modellieren und die Algorithmen und Techniken zur Lösung von LPs und ILPs zeigen hervorragende Ergebnisse bei der Lösung dieser Modelle.</p> <p>Die Vorlesung gibt eine grundlegende Übersicht über lineare und ganzzahlig-lineare Optimierung. Dabei behandeln wir die Modellierung als LP/IP, Algorithmen zur exakten Lösung von LPs / IPs, betrachten effizient lösbare Spezialfälle von IPs und Heuristiken zur Steigerung der Effizienz der Löser im allgemeinen Fall.</p> <p>Stichworte: Simplexalgorithmus, Innere-Punkte-Verfahren, Schnittebenenverfahren, Branch &amp; Bound-Algorithmus, Branch &amp; Cut-Verfahren</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende lernen grundlegende Modellierungstechniken für Optimierungsprobleme. Sie können erkennen, welche Modellierungen für ein Anwendungsproblem geeignet sind und ob sich die erstellten Modelle effizient lösen lassen. Für lineare und ganzzahlige Programme kennen die Studierenden grundlegende Algorithmen zur Lösung sowie unterschiedliche Techniken zur Effizienzsteigerung. Desweiteren können Studierende kommerzielle und nichtkommerzielle Software zur Lösung von LPs und IPs einsetzen.	
<b>Nichtkognitive Kompetenzen</b>	

<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Linear and Integer Optimization: Vorlesung mit Übung
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Rainer Feldmann
<b>Lernmaterialien, Literaturangaben</b>
Linear and Integer Optimization: Folien, Skript Literatur: M. Conforti, G. Cornuejols, G. Zambelli. Integer Programming. Springer 2014. A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency, Vol. A,B,C. Springer 2003. V. Chvatal. Linear Programming. Freeman 1983. L.A. Nemhauser, G.L. Wolsey. Integer and Combinatorial Optimization. Wiley 1999. A. Schrijver. Theory of Linear and Integer Optimization. Wiley, 1999.
<b>Sonstige Hinweise</b>
keine

### 3.30 Wahlpflichtmodul: Logic and Automated Reasoning

Modulname	Logic and Automated Reasoning / Logic and Automated Reasoning
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Logik und automatisches Beweisen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Logik und automatisches Beweisen: Vorlesung ( 45h / 105h / EN / WS / 30 ) Logik und automatisches Beweisen: Übung ( 30h / 0h / EN / WS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Logik und automatisches Beweisen: Grundlagen symbolischer Modellierung, Grundlagen zur Komplexitätstheorie	
<b>Inhalte</b>	
Logik und automatisches Beweisen: Die Veranstaltung "Logik und automatisches Beweisen" soll verschiedene Aspekte der Argumentation und Beweisführung verständlich machen. Basis ist eine symbolische Repräsentation von Wissen über Problemdomänen. Für die Wissensrepräsentation werden logikbasierte Sprachen (klassisch und nicht-klassisch) vorgestellt und ihre Ausdrucksfähigkeit untersucht. Für diese Sprachen werden jeweils passende Deduktionsansätze besprochen und analysiert. Insbesondere werden Zusammenhänge zwischen aussagenlogischen und prädikatenlogischen Ansätzen beleuchtet, sowie spezielle Fragestellungen wie Korrektheit, (Un-) Vollständigkeit und (Nicht-) Monotonie.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage <ul style="list-style-type: none"> <li>• logikbasierte Sprachen zur Wissensrepräsentation zu benennen, ihre Eigenschaften zu beschreiben und zu bewerten,</li> <li>• Wissen und Aufgabenstellungen aus einfachen Domänen mit Hilfe eines logikbasierten Formalismus symbolisch zu repräsentieren,</li> <li>• verschiedene Kalküle und zugehörige Inferenzverfahren zu beschreiben,</li> <li>• geeignete Inferenzverfahren auszuwählen und anzuwenden,</li> <li>• die Komplexität der Aufgabenstellung und Laufzeit von Inferenzverfahren anhand einfacher Skalen zu beurteilen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Logik und automatisches Beweisen: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Übungen</li> <li>• Diskussion</li> <li>• Hausaufgaben</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Theodor Lettmann
<b>Lernmaterialien, Literaturangaben</b>
Logik und automatisches Beweisen: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Lehrbücher: 1.) J. Harrison: Handbook of Practical Logic and Automated Reasoning, Cambridge University Press, 2009, 2.) M. Fitting: First-Order Logic and Automated Theorem Proving, 2nd ed., Springer 1995, 3.) j.H. Gallier: Logic for Computer Science: Foundations of Automatic Theorem Proving, 2nd ed., Dover, 2015, 4.) L. Wos, G. W. Pieper. A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning, World Scientific, 1999.</li> </ul>

- Übungsaufgaben
- Liste von klassischen und aktuellen Papieren, z.B. D. W. Loveland. Mechanical theorem-proving by model elimination, Journal of the ACM,15, pp. 236–251, 1968

**Sonstige Hinweise**

keine



### 3.31 Wahlpflichtmodul: Logic Programming for Artificial Intelligence

Modulname	Logic Programming for Artificial Intelligence / Logic Programming for Artificial Intelligence
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Logic Programming for Artificial Intelligence : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Logic Programming for Artificial Intelligence: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Logic Programming for Artificial Intelligence: Übung ( 30h / 0h / EN / WS oder SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Logic Programming for Artificial Intelligence: Studenten sollten Vorkenntnisse in der Programmierung haben, wie sie in den Lehrveranstaltungen "Programmierung und GPS" angeboten werden, sowie Kenntnisse in Datenbank-Anfragesprachen wie sie in der Lehrveranstaltung "Datenbanksysteme" angeboten werden.	
<b>Inhalte</b>	
Logic Programming for Artificial Intelligence: Diese Lehrveranstaltung betrachtet verschiedene Konzepte und Techniken der Informatik, der Künstlichen Intelligenz und der Computerlinguistik aus einer anderen Perspektive, aus der Perspektive der Logikprogrammierung. Logikprogrammierung im Allgemeinen und die Programmiersprache Prolog im Besonderen erlauben es, viele Konzepte deklarativ in Logik zu beschreiben und gleichzeitig durch einen Interpreter zu testen und auszuführen. Dies eignet sich besonders für die Lösung von Puzzle- und Quiz-Aufgaben, aber auch für selbst definierte und Domänen-spezifische Sprachen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Studenten lernen</p> <p>Faktenwissen über</p> <ul style="list-style-type: none"> <li>• die Transformation von Wissen, das als Fakten und Regeln gegeben ist, in ausführbare Programme</li> <li>• die Programmierung in Logik und in selbst entworfenen Sprachen</li> </ul> <p>Methodisches Wissen, unter anderem die Fähigkeit:</p> <ul style="list-style-type: none"> <li>• Domänen-spezifische Sprachen zu definieren</li> <li>• Interpreter für Domänen-spezifische Sprachen zu implementieren</li> <li>• kleine Frage-Antwort-Systeme zu implementieren</li> </ul>	

<ul style="list-style-type: none"> <li>• Software zu entwickeln für Theorembeweiser, Constraint-Solver und zur Lösung von Puzzles</li> </ul> <p>Transfer-Wissen, u.a. die Fähigkeit:</p> <ul style="list-style-type: none"> <li>• die Methoden auf neue Probleme, Kalküle und Wissensrepräsentationsformate zu übertragen</li> <li>• das Wissen über Parsing und Semantik-Konstruktion auf Domänen-spezifische Sprachen zu übertragen</li> </ul> <p>Normatives Wissen, unter anderem die Fähigkeit folgendes zu bewerten:</p> <ul style="list-style-type: none"> <li>• die Eignung und die Grenzen verschiedener Daten- und Wissensrepräsentationsformate für verschiedene Aufgaben</li> <li>• die Eignung verschiedener Programmier-Paradigmen für verschiedene Projekte</li> <li>• den Aufwand und die Machbarkeit von Projekten mit dem Ziel natürliche Sprache zu verstehen</li> <li>• den Aufwand und die Machbarkeit von Projekten mit dem Ziel natürliche Sprache zu übersetzen</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Logic Programming for Artificial Intelligence: Theoretische Konzepte werden in der Vorlesung erläutert und in den Tutorien in kleinen Gruppen vertieft. Tutorien werden als praktische Übungen am Computer durchgeführt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–

<b>Modulbeauftragte/r</b>
Prof. Dr. Stefan Böttcher
<b>Lernmaterialien, Literaturangaben</b>
Logic Programming for Artificial Intelligence: Ivan Bratko: Prolog Programming for Artificial Intelligence. Pearson Education, Newest Edition. Hinweise auf weiteres Material werden in der Lehrveranstaltung bekannt gegeben.
<b>Sonstige Hinweise</b>
keine

### 3.32 Wahlpflichtmodul: Machine Learning I

Modulname	Machine Learning I / Machine Learning I
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Maschinelles Lernen I : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Maschinelles Lernen I: Vorlesung ( 45h / 105h / EN / WS oder SS / 60 ) Maschinelles Lernen I: Übung ( 30h / 0h / EN / WS oder SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Maschinelles Lernen I: Grundlagen in Mathematik (lineare Algebra, Statistik), Programmierung und Algorithmen.	
<b>Inhalte</b>	
Maschinelles Lernen I: Aufgrund der stetig wachsenden Menge an Daten, die in unserer Informationsgesellschaft systematisch produziert wird, hat das Maschinelle Lernen in den letzten Jahren mehr und mehr an Bedeutung gewonnen, nicht nur als wissenschaftliche Disziplin sondern auch als Schlüsseltechnologie für moderne Software und intelligente Systeme. Diese Vorlesung gibt eine Einführung in das Maschinelle Lernen, wobei der Fokus auf dem überwachten Lernen für Klassifikation und Regression liegt. Theoretische Grundlagen der Generalisierung werden ebenso behandelt wie praktische Aspekte und konkrete Lernalgorithmen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden verstehen die statistischen Grundlagen der Generalisierung, d.h. der Induktion von Modellen aus Daten, sowie praktischen Ansätzen zur Modellvalidierung. Sie können grundlegende Methoden und Algorithmen des überwachten Lernens auf Klassifikations- und Regressionsprobleme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	

Maschinelles Lernen I: Theoretische Grundlagen und Konzepte des Maschinellen Lernens werden im Rahmen einer Vorlesung eingeführt und anschließend in praktischen Übungen in Kleingruppen sowie in Heimübungen vertieft ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eyke Hüllermeier
<b>Lernmaterialien, Literaturangaben</b>
Maschinelles Lernen I: Skript und eine Liste mit Buchempfehlungen: <ul style="list-style-type: none"> <li>• Y.S. Abu-Mostafa, M. Magdon-Ismael, H.T. Lin. Learning from Data, AMLBook, 2012.</li> <li>• P. Flach. Machine Learning, Cambridge Univ. Press, 2012.</li> <li>• E. Alpaydin. Machine Learning, Oldenbourg, 2008.</li> <li>• C.M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.33 Wahlpflichtmodul: Machine Learning II

Modulname	Machine Learning II / Machine Learning II
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Maschinelles Lernen II : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Maschinelles Lernen II: Vorlesung ( 45h / 105h / EN / WS oder SS / 20 ) Maschinelles Lernen II: Übung ( 30h / 0h / EN / WS oder SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Maschinelles Lernen II: Grundlegende Kenntnisse in Maschinellem Lernen (z.B. vermittelt durch die Machine Learning I Vorlesung).	
<b>Inhalte</b>	
Maschinelles Lernen II: Aufbauend auf einer grundlegenden Einführung in das maschinelle Lernen, wie beispielsweise vermittelt durch die Veranstaltung Machine Learning I, werden in dieser Vorlesung fortgeschrittene Themen in diesem Gebiet behandelt (reinforcement learning, online learning and bandit algorithms, multi-task learning, multi-target and structured output prediction, preference learning, learning from weak supervision, and uncertainty in machine learning). Obwohl die Vorlesung im Wesentlichen methodisch und algorithmisch ausgerichtet ist, werden auch theoretische und anwendungsorientierte Aspekte behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden haben einen Überblick über Methoden zur Klassifikation mit mehr als zwei Klassen, dem Lernen nichtlinearer Modelle, sowie Erweiterungen des einfachen Szenarios des überwachten Lernens. Sie verstehen algorithmische Konzepte entsprechender Lernverfahren und können diese Verfahren auf praktische Probleme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	

Maschinelles Lernen II: Theoretische Grundlagen und Konzepte des Maschinellen Lernens werden im Rahmen einer Vorlesung eingeführt und anschließend in praktischen Übungen in Kleingruppen sowie in Heimübungen vertieft ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eyke Hüllermeier
<b>Lernmaterialien, Literaturangaben</b>
Maschinelles Lernen II: Skript und eine Liste mit Buchempfehlungen: <ul style="list-style-type: none"> <li>• Y.S. Abu-Mostafa, M. Magdon-Ismael, H.T. Lin. Learning from Data, AMLBook, 2012.</li> <li>• P. Flach. Machine Learning, Cambridge Univ. Press, 2012.</li> <li>• E. Alpaydin. Machine Learning, Oldenbourg, 2008.</li> <li>• C.M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.34 Pflichtmodul: Master-Abschlussarbeit

Modulname	Master-Abschlussarbeit / Master Thesis
Workload	900 h
Leistungspunkte	30 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Masterarbeit Arbeitsplan : 4</li> <li>• Masterarbeit : 4</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Masterarbeit Arbeitsplan: Kontaktzeiten, Präsentation Arbeitsplan ( 30h / 120 h / EN / WS oder SS / 0 ) Masterarbeit: Kontaktzeiten, Ergebnispräsentation ( 30h / 720 h / EN / WS oder SS / 1 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Modulprüfungen im Hauptfach im Umfang von 48 LP müssen erfolgreich abgelegt worden sein.	
<b>Empfohlene Kenntnisse</b>	
Masterarbeit Arbeitsplan: Je nach gewähltem Thema. Masterarbeit: Je nach gewähltem Thema.	
<b>Inhalte</b>	
Masterarbeit Arbeitsplan: Nach Themenabsprache mit dem Betreuer erfolgt eine erste grobe Einarbeitung. Auf dieser Grundlage und einer ersten Literaturrecherche ist durch den Studierenden ein Arbeitsplan vorzulegen, der die zu erzielenden Ergebnisse samt Meilensteine für die Arbeit dokumentiert.  Masterarbeit: In der Masterarbeit zeigt der/die Kandidat/in seine/ihre Fähigkeit zum selbständigen wissenschaftlichen Arbeiten an einem angemessen anspruchsvollen Thema, das auch Gelegenheit zur Entfaltung eigener Ideen gibt. Auf der Grundlage des State-of-the-art sollen die Methoden der Informatik systematisch angewendet werden. Die Ergebnisse der Arbeit werden in einem fakultätsöffentlichen Vortrag vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Mit Abschluss der Masterarbeit haben die Studierenden gezeigt, dass sie <ul style="list-style-type: none"> <li>• ein Problem innerhalb einer bestimmten Frist ein Problem nach wissenschaftlichen Methoden bearbeiten können,</li> <li>• die im Zuge des Studiums erworbenen fachlich-methodischen fachübergreifenden Kompetenzen zur Lösung komplexer Aufgabenstellungen anwenden können. Die Schwerpunkte variieren ja nach Aufgabenstellung.</li> </ul>	



<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Masterarbeit Arbeitsplan: Direkte Absprache mit Betreuer.</p> <p>Masterarbeit: Selbständiges Arbeiten unterstützt durch individuelle Betreuung</p>
<b>Prüfungsleistung (Dauer)</b>
<p>Abschlussarbeit</p> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 50 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
<p>Masterarbeit Arbeitsplan: Je nach gewählttem Thema.</p> <p>Masterarbeit: Je nach gewählttem Thema.</p>
<b>Sonstige Hinweise</b>
Die Masterarbeit ist eine Prüfungsarbeit, die die wissenschaftliche Ausbildung abschließt und zeigen soll, dass die Kandidatin oder der Kandidat die Fähigkeit besitzt, innerhalb einer bestimmten Frist ein

Problem der Informatik nach wissenschaftlichen Methoden zu bearbeiten. Die Aufgabenstellung soll so gestaltet werden, dass sie einem Arbeitsaufwand von fünf Monaten Vollzeitarbeit entspricht. Die Arbeit muss fünf Monate nach der Ausgabe abgegeben werden. Die Arbeit soll einen Umfang von in der Regel nicht mehr als 120 DIN A4-Seiten haben.

### 3.35 Wahlpflichtmodul: Mobile Communication

Modulname	Mobile Communication / Mobile Communication
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Mobilkommunikation : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Mobilkommunikation: Vorlesung ( 45h / 105h / EN / WS / 20 ) Mobilkommunikation: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Mobilkommunikation: Grundkenntnisse in Rechnernetze (z.B. durch die Bachelor-Vorlesung Rechnernetze).	
<b>Inhalte</b>	
Mobilkommunikation: Die Veranstaltung behandelt grundlegende Techniken für die Mobilkommunikation (z.B. drahtlose Kanalmodelle) und Techniken (z.B. Spreizbandkommunikation), wesentliche Protokollmechanismen (z.B. Medienzugriff), Systeme der Mobilkommunikation sowie MobileIP. Neben technologischen und konzeptionellen Aspekten werden auch Verfahren und Methoden zur Leistungsbewertung besprochen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer kennen die Herausforderungen und Probleme beim Entwurf und Betrieb von Mobilkommunikationssystemen. Sie können zwischen physikalischen und entwurfsbedingten Problemstellungen differenzieren und geeignete Protokollmuster auswählen bzw. neue Protokolle konstruieren. Sie sind in der Lage, Mechanismen unterschiedlicher Architekturebenen auszuwählen, in eine sinnvolle Gesamtarchitektur zu integrieren und diese Auswahl zu begründen. Sie sind in der Lage, Protokollmechanismen quantitativ zu evaluieren (was auch fachübergreifend einsetzbar ist).	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	

Mobilkommunikation: Vorlesung mit Folien und Tafelanschrieb; begleitende Übungen u.a. mit Programmieraufgaben zu einfachen Simulationen drahtloser Systeme.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Mobilkommunikation: Foliensatz; einzelne Kapitel div. Standardlehrbücher (J. Schiller, Mobile Communication, Addison Wesley, 2nd edition; D. Tse und P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005).
<b>Sonstige Hinweise</b>
keine

### 3.36 Wahlpflichtmodul: Model Checking

Modulname	Model Checking / Model Checking
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Model Checking : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Model Checking: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Model Checking: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Model Checking: Logic, imperative Programmierung, Parallelität	
<b>Inhalte</b>	
Model Checking: In dieser Vorlesung werden automatische Verfahren zur Verifikation von Software- oder Hardwaresystemen vorgestellt, d.h. Verfahren, die prüfen, ob ein System korrekt gegenüber spezifizierten Anforderungen ist. Für die Beschreibung der Anforderungen werden dabei temporale Logiken (CTL und LTL) genutzt. Solche Verifikationsverfahren sind in Werkzeugen umgesetzt, von denen eines in der Vorlesung erklärt und in den Übungen genutzt wird.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage für Software oder Hardwaresysteme Anforderungen formal in temporaler Logik zu spezifizieren und kennen Werkzeuge, um diese Anforderungen automatisch zu prüfen. Sie kennen die Funktionsweise von Modelchecking-Algorithmen, und die Unterschiede zwischen linear-time und branching-time Logiken. Sie können neue Forschungsansätze im Bereich Verifikation verstehen und in Bestehendes einordnen. Die Studierenden können einfache formale Beweise selber entwickeln sowie die in der Vorlesung vorgestellten Beweise erklären.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	
Model Checking: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. In praktischen Übungen mit dem Modelcheck-	

er Spin werden die erlernten Kenntnisse angewendet.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Model Checking: Christel Baier, Joost-Pieter Katoen: Principles of Model Checking Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.37 Wahlpflichtmodul: Model-Based Interface Development

Modulname	Model-Based Interface Development / Model-Based Interface Development
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Modellbasierte Entwicklung von Benutzungsschnittstellen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Modellbasierte Entwicklung von Benutzungsschnittstellen: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Modellbasierte Entwicklung von Benutzungsschnittstellen: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Modellbasierte Entwicklung von Benutzungsschnittstellen: Die Vorlesung setzt keine besonderen Vorkenntnisse, die über das vorangegangene Bachelorstudium hinausgehen, voraus.	
<b>Inhalte</b>	
<p>Modellbasierte Entwicklung von Benutzungsschnittstellen: Die Vorlesung beginnt mit einer klärenden Einführung des Modellbegriffs innerhalb der Informatik. Modelle werden hier wie folgt definiert: Ein Modell ist ein Artefakt, wie zum Beispiel eine Zeichnung oder ein physisches Objekt, das einen bestimmten wichtigen Aspekt eines realen Systems wiedergibt oder repräsentiert. Ausgehend von dieser Definition werden eine Reihe von Typen von Artefakten zur Abdeckung verschiedener Aspekte betrachtet. Es beginnt mit Ansätzen zur Aufgabenmodellierung, wo wir verschiedene Methoden und Werkzeuge - mit denen auch gearbeitet wird - betrachten. Danach behandeln wir Dialogmodelle, die in zwei grundverschiedene Gruppen aufgeteilt werden: Kontrollmodelle und Benutzerinteraktionsmodelle. Beide Typen beschreiben eigentlich dieselbe Sache", haben aber verschiedene Ziele und führen zu verschiedenen Ergebnissen. Wir behandeln pro Typ mehrere Ansätze und ihre Querbezüge, wann immer möglich mit Betrachtung entsprechender Werkzeuge. Betrachtet wird auch ein typischer modellbasierter Benutzungsschnittstellen-Entwicklungsprozess. Die gefundenen Erkenntnisse werden anschließend auf den Bereich der Webmodellierung übertragen. Die Probleme sind im wesentlichen die gleichen wie bei der klassischen Benutzungsschnittstelle, allerdings werden andere, dem Bereich angepassten, Modellierungsansätze verwendet. Wir betrachten Problembereichsmodell, Navigationsmodell und Präsentationsmodell für Webauftritte. Neben (noch) rein akademisch interessanten Ansätzen wird intensiv die existierende, lauffähige Webmodellierungsumgebung WebRatio behandelt und praktisch eingesetzt.</p>	

<b>Lernergebnisse / Fachkompetenzen</b>
Faktenwissen: Die Vorlesung vermittelt explizit Kalküle, die beim Modellieren zum Einsatz kommen. Dadurch wird auch deutlich gemacht, welche inhaltlichen Aspekte, die bei der Entwicklung von Benutzungsschnittstellen eine Rolle spielen, mit welchem Modell und auf welcher Abstraktionsebene behandelt werden können. Es werden sowohl die Grundlagen für diese Betrachtungen behandelt, als auch spezielle, für das Web relevante Aspekte und Modellierungskonzepte. Methodisches Wissen: Die Studierenden lernen die behandelten Modellierungstechniken einzusetzen. Es werden teils akademische, teils kommerzielle Modellierungswerkzeuge vorgestellt und ihr Gebrauch in den Übungen erprobt. Transferkompetenz: Die Studierenden lernen wie formale Kalküle in realen Entwurfsszenarien hilfreich eingesetzt werden können. Normativ-bewertend: Die Studierenden lernen den Nutzen, aber auch die Defizite der konkreten Modellierungsansätze, aber auch des modellbasierten Ansatzes an sich kennen. Sie können im Ergebnis bei der Lösung praktischer Aufgaben beurteilen, ob der Einsatz einer modellbasierten Vorgehensweise angebracht ist und welche Vorteile und Risiken damit verbunden sind.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Modellbasierte Entwicklung von Benutzungsschnittstellen: Die Vorlesung wird klassisch mit Beamerpräsentation gehalten und intensiv durch die E-Learning-Umgebung koaLA der Universität Paderborn unterstützt. Hier werden vor der Vorlesung die Folien veröffentlicht, schriftliche Übungsaufgaben gestellt, Software (Modellierungswerkzeuge) und Videoaufzeichnungen aller Vorlesungen zur Verfügung gestellt. Im Rahmen der Vorlesung selbst findet immer wieder an geeigneter Stelle interaktive Gruppenarbeit statt – etwa zum Modellieren mit WebRatio oder der Erstellung komplexer Aufgabenmodelle. In den Übungsgruppen stellen Hörer die von ihnen erarbeiteten Lösungen vor und zur Diskussion. Dazu müssen die Vortragenden ihre Lösung anhand von geeigneten Vortragsfolien präsentieren.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>



Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gerd Szwillus
<b>Lernmaterialien, Literaturangaben</b>
<p>Modellbasierte Entwicklung von Benutzungsschnittstellen:</p> <ul style="list-style-type: none"> <li>• Vorlesungsfolien als PDF zum Herunterladen</li> <li>• Schriftliche Hausaufgaben</li> <li>• Diverse, während der Vorlesung eingesetzte Modellierungswerkzeuge</li> <li>• Voller Zugang zu WebRatio</li> <li>• Ben Shneiderman: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2009</li> <li>• David Benyon: Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design, 2009</li> <li>• Stefano Ceri et al: Designing Data-Intensive Web Applications, 2003</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.38 Wahlpflichtmodul: Network Simulation

Modulname	Network Simulation / Network Simulation
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Network Simulation : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Network Simulation: Vorlesung ( 30h / 105h / EN / SS / 20 ) Network Simulation: Übung ( 45h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Network Simulation: Systemsoftware und systemnahe Programmierung	
<b>Inhalte</b>	
<p>Network Simulation: Der Kurs fokussiert auf dem Simulationswerkzeug OMNeT++. Simulation ist eine der grundlegenden Möglichkeiten (neben Experimenten und mathematischer Analyse), die Leistung von Systemen zu bewerten, auch wenn diese noch nicht in als reales System verfügbar sind.</p> <p>Nach eine grundlegenden Einführung in Simulatio und Modellierung werden anhand von kleinen Beispielprojekten die Möglichkeiten von OMNeT++ exploriert. Letztendlich soll aber ein forschungsrelevantes Projekt in Gruppenarbeit (2-3 Studierende pro Gruppe) durchgeführt werden. Die Themen kommen aus den Gebieten Fahrzeugkommunikation und Sensornetze.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Lernziel ist es, grundlegende Konzepte der Netzwerksimulation zu studieren. Die Studierenden verstehen diese Konzepte und sind in der Lage, dieses Wissen anzuwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Network Simulation: Vorlesung mit praktischen Übungen	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Network Simulation: Folien, Lehrbücher, Papiere
<b>Sonstige Hinweise</b>
keine

### 3.39 Wahlpflichtmodul: Networked Embedded Systems

Modulname	Networked Embedded Systems / Networked Embedded Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Networked Embedded Systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Networked Embedded Systems: Vorlesung ( 45h / 105h / EN / WS / 60 ) Networked Embedded Systems: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Networked Embedded Systems: Systemsoftware und systemnahe Programmierung	
<b>Inhalte</b>	
Networked Embedded Systems: Ziel des Kurses ist es, vertiefte Einblicke in den Entwurf und die Programmierung eingebetteter Systeme zu erlangen. Der Fokus liegt klar auf der Anwendungsdomäne Sensornetze. Daher werden fundamentale Grundlagen von Sensornetzen untersucht und im Rahmen der Übungen vertieft.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Ziel ist es, grundlegende Konzepte vernetzter eingebetteter Systeme zu verstehen. Die Studierenden sind in der Lage, diese Konzepte anzuwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Networked Embedded Systems: Vorlesung mit praktischen Übungen	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.	

<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Networked Embedded Systems: Folien, Lehrbücher, Papiere
<b>Sonstige Hinweise</b>
keine

### 3.40 Wahlpflichtmodul: Planning and Heuristic Search

Modulname	Planning and Heuristic Search / Planning and Heuristic Search
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Planen und Heuristische Suche : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Planen und Heuristische Suche: Vorlesung ( 45h / 105h / EN / WS / 40 ) Planen und Heuristische Suche: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Planen und Heuristische Suche: Grundlagen symbolischer Modellierung: Funktionen, Relationen, logische Formeln; Design und Analyse von Algorithmen; Grundlagen zur Komplexitätstheorie: Komplexitätsklassen, Reduzierbarkeit, Vollständigkeit.	
<b>Inhalte</b>	
Planen und Heuristische Suche: Die Veranstaltung Planen und Heuristische Suche stellt zwei Ansätze zum Lösen wissensintensiver Aufgaben vor. Im Bereich Planen werden Repräsentationen von Aufgabenstellungen als Planungsproblem in Zustandsräumen oder Planräumen vorgestellt und passende Verfahren diskutiert und analysiert. Im Bereich Heuristische Suche wird das Konzept des Zustandsraums verallgemeinert und ein Programmrahmen für systematische Suchverfahren beschrieben, der es erlaubt, die Suche durch Nutzung heuristischer Informationen über die Problemdomäne zu fokussieren. Als ein Anwendungsbeispiel werden Planungsverfahren als heuristische Suche implementiert. In beiden Bereichen werden theoretische Ergebnisse vorgestellt und bewiesen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage, <ul style="list-style-type: none"> <li>• die Konzepte zur Modellierung von Planungs- und Suchaufgaben zu benennen und zu erklären,</li> <li>• die Vorgehensweisen von verschiedenen Planungsverfahren sowie von heuristischen Suchverfahren zu beschreiben,</li> <li>• einfache Aufgabenstellungen als Planungs- bzw. Suchaufgaben zu erkennen und zu repräsentieren,</li> <li>• darin Ansätze zur Erstellung brauchbarer Heuristiken zu entdecken,</li> <li>• theoretische Ergebnisse als Hinweise für die Auswahl von Modellierung und Verfahren zu verwenden.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> </ul>	

<ul style="list-style-type: none"> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Planen und Heuristische Suche: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Übungen</li> <li>• Diskussion</li> <li>• Hausaufgaben</li> <li>• Referenzimplementationen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Theodor Lettmann
<b>Lernmaterialien, Literaturangaben</b>
Planen und Heuristische Suche: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Lehrbücher: 1.) J. Pearl: Heuristics, Addison-Wesley (1984), 2.) S.J. Russell, P. Norvig: Artificial Intelligence: A Modern Approach, Prentice Hall, 1995, 3.) S. Edelkamp, S. Schrödl: Heuristic Search: Theory and Applications, Elsevier, 2012, 4.) M. Ghallab, D. Nau, P. Traverso: Automated Planning,</li> </ul>

Morgan Kaufmann, 2004

- Übungsaufgaben
- Liste von klassischen und aktuellen Papieren, z.B. R. Eberdt, R. Drechsler: Weighted A\*Search - Unifying View and Application, J. Artificial Intelligende, pp. 1310-1342, 2009 item Online Material, z.B. H. Geffner, B. Bonet: A Concise Introduction to Models and Methods for Automated Planning, doi: 10.2200/S00513ED1V01Y201306AIM022, 2013

**Sonstige Hinweise**

keine



### 3.41 Pflichtmodul: Projektgruppe

Modulname	Projektgruppe / Proje Group
Workload	600 h
Leistungspunkte	20 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Projektgruppe : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Projektgruppe: Vorlesung ( 30h / 540h / EN / WS oder SS / 10 ) Projektgruppe: Besprechungen, Präsentation von Teilergebnissen ( 30h / 0 h / EN / WS oder SS / 10 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Projektgruppe: Abhängig vom Thema.	
<b>Inhalte</b>	
Projektgruppe: In einer Projektgruppe bearbeitet eine Gruppe von in der Regel 8-16 Studierenden über den Zeitraum eines Jahres (zwei Semester) ein vom Veranstalter vorgegebenes Thema. Inhaltlich sollen Projektgruppen die Studierenden an aktuelle Forschungsthemen heranzuführen und durch die Teamarbeit auf die Arbeitsweise der industriellen Praxis vorbereiten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
In der Projektgruppe wird Teamarbeit und Organisation eines Projekts praktisch erprobt und erlernt; hierdurch werden die Teilnehmer auf die spätere industrielle Berufspraxis vorbereitet. Die Studierenden lernen umfangreiche Entwicklungsprozesse im Team aus eigener Anschauung kennen. Durch die ausdrückliche Arbeitsteilung entsteht der Zwang, über eigene Arbeiten innerhalb der Gruppe zu berichten und die Ergebnisse zu vertreten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
<p>Projektgruppe:</p> <ul style="list-style-type: none"> <li>• Die Zahl der Projektgruppenteilnehmer ist auf 16 Personen begrenzt.</li> <li>• Es finden Plenumstreffen (alle Teilnehmer und der Veranstalter) statt, insbesondere zur Vermittlung gemeinsam erforderlichen Wissens (Seminarphasen zu Beginn der beiden Semester) und zur Planung der weiteren Arbeit.</li> <li>• Es werden feste Verantwortlichkeiten zwischen den Teilnehmern aufgeteilt, die über die gesamte Projektlaufzeit oder auch nur kurzfristig (ad-hoc-Aufgaben) Bestand haben können.</li> <li>• Es werden Untergruppen zu einzelnen Themen gebildet, die selbständig und termingebunden Aufgaben vorantreiben und dem Plenum Rechenschaft ablegen müssen.</li> <li>• Typischerweise erarbeitet jede Projektgruppe auch eine Repräsentation ihrer Arbeit in einer Webseite.</li> <li>• Am Ende jedes der beiden Semester ist ein Bericht zu erstellen, der in jedem Aspekt von den Teilnehmern gestaltet und mit Inhalt gefüllt wird.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Phasenbezogene Prüfung (100 % der Modulnote)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: Praktikumsarbeit  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 8 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Projektgruppe: Abhängig vom Thema.

**Sonstige Hinweise**

Im Modul Projektgruppe ist die erfolgreiche Bearbeitung von Projekten durch die Abgabe von Software und Dokumentation als phasenbezogene Prüfung nachzuweisen. Es wird eine Note für die Gesamtheit der bearbeiteten Projekte vergeben.

### 3.42 Wahlpflichtmodul: Public-Key Cryptography

Modulname	Public-Key Cryptography / Public-Key Cryptography
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Public-Key Cryptography : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Public-Key Cryptography: Vorlesung ( 45h / 105h / EN / WS / 50 ) Public-Key Cryptography: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Public-Key Cryptography: Elementare Grundlagen der Kryptographie, wie sie in einer Vorlesung wie "Einführung in die Kryptographie" vermittelt werden.	
<b>Inhalte</b>	
Public-Key Cryptography: Diese Veranstaltung vermittelt grundlegende Techniken zur Konstruktion und formalen Sicherheitsanalyse von Public-Key Kryptosystemen. Dies umfasst insbesondere digitale Signaturen, Public-Key Verschlüsselung, identitätsbasierte Verschlüsselung und ggfs. weitere verwandte Konzepte. Der Hauptfokus liegt hierbei auf der Vorstellung der Techniken, die modernen Kryptosystemen und deren Sicherheitsbeweisen zugrunde liegen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende kennen etablierte Techniken zur Konstruktion beweisbar sicherer kryptographischer Public-Key Verfahren. Sie sind in der Lage Sicherheitsanforderungen formal zu definieren, kennen die etablierten Standard-Definitionen und ihre Grenzen und sind in der Lage die Beziehungen zwischen diesen Definitionen formal zu analysieren. Sie können Standard-Beweistechniken auch auf neue Kryptosysteme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Einsatz und Engagement</li> <li>Lernkompetenz</li> <li>Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Public-Key Cryptography: <ul style="list-style-type: none"> <li>Vorlesung mit Beamer und Tafelanschrieb</li> </ul>	

<ul style="list-style-type: none"> <li>• Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr., Tibor Jager
<b>Lernmaterialien, Literaturangaben</b>
Public-Key Cryptography: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Tafelanschrieb</li> <li>• Originalliteratur</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.43 Wahlpflichtmodul: Reconfigurable Computing

Modulname	Reconfigurable Computing / Reconfigurable Computing
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Reconfigurable Computing : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Reconfigurable Computing: Vorlesung ( 30h / 105h / EN / WS / 50 ) Reconfigurable Computing: Übung ( 45h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Reconfigurable Computing: Kenntnisse aus Digitaltechnik und Rechnerarchitektur sind hilfreich.	
<b>Inhalte</b>	
Reconfigurable Computing: Die Veranstaltung vermittelt Kenntnisse in Architekturen und Entwurfsmethoden für rekonfigurierbare Hardware und stellt Anwendungen im Bereich des Hochleistungsrechnens und der eingebetteten Systeme vor.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• den Aufbau rekonfigurierbarer Hardwarebausteine zu erklären,</li> <li>• die wesentlichen Entwurfsmethoden zu benennen und zu analysieren und</li> <li>• die Eignung rekonfigurierbarer Hardware für verschiedene Einsatzgebiete zu beurteilen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Reconfigurable Computing: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Rechnerübungen mit rekonfigurierbaren Systemen</li> </ul>	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Reconfigurable Computing: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• S. Hauck and A. DeHon (editors): Reconfigurable Computing, Volume 1: The Theory and Practice of FPGA-Based Computation, Morgan Kaufmann, 2008</li> <li>• Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.44 Wahlpflichtmodul: Routing and Data Management in Networks

Modulname	Routing and Data Management in Networks / Routing and Data Management in Networks
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Routing and Data Management in Networks : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Routing and Data Management in Networks: Vorlesung ( 45h / 105h / EN / SS / 40 ) Routing and Data Management in Networks: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Routing and Data Management in Networks: Algorithmen-Entwurf, theoretische Korrektheit und Effizienzbeweise, Werkzeuge aus der Kombinatorik und Wahrscheinlichkeitstheorie.	
<b>Inhalte</b>	
Routing and Data Management in Networks: Routing und Datenmanagement sind grundlegenden zu lösende Aufgaben, um eine effiziente Verwendung von großen Netzwerken wie z.B. dem Internet, Peer-to-Peer-Systemen, oder drahtlosen mobilen Ad-hoc-Netzwerke zu ermöglichen. Diese Vorlesung befasst sich mit Algorithmen und deren Analyse für das Routing und Datenmanagement in solchen Systemen und beschreibt insbesondere Methoden für den Umgang mit ihrer Dynamik (Bewegung von Knoten, Beitritt und Austritt von Knoten). Dabei werden insbesondere lokale, verteilte Algorithmen, häufig als online Algorithmen betrachtet.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden lernen die wichtigsten Techniken im Bereich des Routing und Datenmanagements von großen Netzwerken kennen. Sie können entscheiden, in welcher Situation welcher Datenmanagement-, Scheduling- oder Routing-Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	



<b>Methodische Umsetzung</b>
Routing and Data Management in Networks: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Bearbeitung der Hausaufgaben, Mitarbeit in den Übungen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Friedhelm Meyer auf der Heide
<b>Lernmaterialien, Literaturangaben</b>
Routing and Data Management in Networks: Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Frank Thomson Leighton, M. Kaufmann Publishers, 1992. Originalarbeiten, Skript, Foliensatz der Vorlesung, Übungsblätter
<b>Sonstige Hinweise</b>
keine

### 3.45 Pflichtmodul: Seminar I

Modulname	Seminar I / Seminar I
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Seminar : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Seminar: Seminar ( 30h / 120 h / EN / WS oder SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Seminare aus dem Masterstudiengang Informatik.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Seminar: Abhängig vom Seminarthema.	
<b>Inhalte</b>	
<p>Seminar: In Seminaren erarbeiten sich die Teilnehmer ein Thema, welches in einem Vortrag mit anschließender Diskussion und einer schriftlichen Ausarbeitung präsentiert wird. Vortragsmaterial und Ausarbeitung dienen dabei unterschiedlichen Zielen: Während das Vortragsmaterial zur Unterstützung des Vortrags dient (der in engen zeitlichen Grenzen abläuft), dient die Ausarbeitung dazu, sich zu einem späteren Zeitpunkt detailliert über das Thema informieren zu können.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Im Seminar sollen die Studierenden Techniken zur selbständigen Erarbeitung von nichttrivialem Stoff erlernen, indem sie sich in ein forschungsnahes Teilgebiet der Informatik einarbeiten. Sie sollen lernen, einen Vortrag zu planen, der sich an zeitliche Vorgaben (üblicherweise 45 bis 60 Minuten) hält, und dabei inhaltliche Prioritäten zu setzen. Die Teilnehmer sollen praktisch erfahren, wie man als Zuschauer aus einem Vortrag Kenntnisse aufnimmt, und in Diskussionen Meinungen und Information austauschen. Seminare dienen auch der Vermittlung rhetorischer Fähigkeiten bei Vortrag und Diskussion. Die Teilnehmer sollen lernen, den Vortrag entlang einer inhaltlichen Linie zu strukturieren und verschiedene Mittel zur Illustration komplexer Sachverhalte zu nutzen. Ebenso soll der angemessene Umgang mit Literatur gelernt werden.</p>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Medienkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Seminar: Referate mit schriftlicher Ausarbeitung und Vortrag.
<b>Prüfungsleistung (Dauer)</b>
Seminarvortrag (45-60 Minuten) und schriftliche Ausarbeitung Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Seminar: Abhängig vom Seminarthema.
<b>Sonstige Hinweise</b>
keine

### 3.46 Pflichtmodul: Seminar II

Modulname	Seminar II / Seminar II
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Seminar : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Seminar: Seminar ( 30h / 120 h / EN / WS oder SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Seminare aus dem Masterstudiengang Informatik.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Seminar: Abhängig vom Seminarthema.	
<b>Inhalte</b>	
Seminar: In Seminaren erarbeiten sich die Teilnehmer ein Thema, welches in einem Vortrag mit anschließender Diskussion und einer schriftlichen Ausarbeitung präsentiert wird. Vortragsmaterial und Ausarbeitung dienen dabei unterschiedlichen Zielen: Während das Vortragsmaterial zur Unterstützung des Vortrags dient (der in engen zeitlichen Grenzen abläuft), dient die Ausarbeitung dazu, sich zu einem späteren Zeitpunkt detailliert über das Thema informieren zu können.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Im Seminar sollen die Studierenden Techniken zur selbständigen Erarbeitung von nichttrivialem Stoff erlernen, indem sie sich in ein forschungsnahes Teilgebiet der Informatik einarbeiten. Sie sollen lernen, einen Vortrag zu planen, der sich an zeitliche Vorgaben (üblicherweise 45 bis 60 Minuten) hält, und dabei inhaltliche Prioritäten zu setzen. Die Teilnehmer sollen praktisch erfahren, wie man als Zuschauer aus einem Vortrag Kenntnisse aufnimmt, und in Diskussionen Meinungen und Information austauschen. Seminare dienen auch der Vermittlung rhetorischer Fähigkeiten bei Vortrag und Diskussion. Die Teilnehmer sollen lernen, den Vortrag entlang einer inhaltlichen Linie zu strukturieren und verschiedene Mittel zur Illustration komplexer Sachverhalte zu nutzen. Ebenso soll der angemessene Umgang mit Literatur gelernt werden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Medienkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Seminar: Referate mit schriftlicher Ausarbeitung und Vortrag.
<b>Prüfungsleistung (Dauer)</b>
Seminarvortrag (45-60 Minuten) und schriftliche Ausarbeitung Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Seminar: Abhängig vom Seminarthema.
<b>Sonstige Hinweise</b>
keine

### 3.47 Wahlpflichtmodul: Software Analysis

Modulname	Software Analysis / Software Analysis
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Software Analysis : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Software Analysis: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Software Analysis: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Software Analysis: Logic, imperative programming	
<b>Inhalte</b>	
Software Analysis: In der Vorlesung werden statische Analysemethoden für Programm vorgestellt. Die Information, die dabei gewonnen wird, kann für die Optimierungsphase von Compilern, aber auch für den Nachweis der Korrektheit von Programmen genutzt werden.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden kennen die wesentlichen Prinzipien von Datenflussanalysen und ihre Vor- und Nachteile. Sie können einschätzen, wann Datenflussanalysen sinnvoll und wann pfadsensitive Analysen benötigt sind. Studierende können eigene Analysen entwerfen und wissen, wie diese umzusetzen sind. Sie kennen das Prinzip der Überapproximation.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	
Software Analysis: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei	

Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Software Analysis: Nielson, Nielson, Hankin: Principles of Program Analysis Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.48 Wahlpflichtmodul: Software Quality Assurance

Modulname	Software Quality Assurance / Software Quality Assurance
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Software Quality Assurance : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Software Quality Assurance: Vorlesung ( 45h / 105h / EN / SS / 90 ) Software Quality Assurance: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Software Quality Assurance: Programmierung, Modellierung, Modellbasierte Softwareentwicklung	
<b>Inhalte</b>	
Software Quality Assurance: Das Ziel der Vorlesung ist die Behandlung von Ansätzen, Technologien und Strategien für die Qualitätssicherung von Softwaresystemen. Dies beinhaltet einerseits konstruktive Ansätze wie Design Pattern, Anti-Pattern, domänenspezifische Sprachen, modellgetriebene Softwareentwicklung, Qualitätsmodelle und Architekturstile und andererseits analytische Ansätze wie statische Reviewtechniken und dynamisch Testtechniken. Des Weiteren werden Ansätze für die Verbesserung des Softwareentwicklungsprozesses und internationale Standards wie ISO 9001, 9126, CMM, usw. behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage, Qualitätseigenschaften von Entwicklungsprozessen, Softwaremodellen bzw. -systemen zu benennen. Sie kennen Techniken zur konstruktiven bzw. analytischen Sicherstellung von Qualitätseigenschaften und können diese geeignet einsetzen. Sie kennen die wesentlichen Standards für die Bewertung von Prozess- und Produktqualitäten. Sie können ausgewählte, aktuelle Forschungsansätze im Bereich Prozess- und Softwarequalitätssicherung verstehen und einordnen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Empathie</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	



<b>Methodische Umsetzung</b>
Software Quality Assurance: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. In praktischen Übungen insbesondere mit Testwerkzeugen werden die erlernten Kenntnisse angewendet.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gregor Engels
<b>Lernmaterialien, Literaturangaben</b>
Software Quality Assurance: Daniel Galin: Software Quality Assurance: From Theory to Implementation, Pearson / Addison Wesley, 2004 Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.49 Pflichtmodul: Studium Generale

Modulname	Studium Generale / General Studies
Workload	360 h
Leistungspunkte	12 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Studium Generale : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Studium Generale: Vorlesung ( 90h / 225h / DE / WS / 30 ) Studium Generale: Übung ( 45h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Beliebige Veranstaltungen außerhalb der Informatik können gewählt werden.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Inhalte</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden erweitern ihren wissenschaftlichen Horizont über die Grenzen der Informatik und des gewählten Nebenfaches hinaus. Je nach gewählter Veranstaltung haben sie Kompetenzen im Bereich Kommunikationsfähigkeit, Teamarbeit und Präsentationstechniken erworben.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Medienkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Prüfungsleistung (Dauer)</b>	
Prüfung im Studium Generale Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.	

<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Qualifizierte Teilnahme: Qualifizierte Teilnahme im Studium Generale Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 4 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dozenten der Universität Paderborn
<b>Lernmaterialien, Literaturangaben</b>
Studium Generale: Abhängig von den gewählten Veranstaltungen.
<b>Sonstige Hinweise</b>
Ist kein Nebenfach gewählt, muss eine beliebige Kombination von Veranstaltungen außerhalb der Informatik und im Umfang von 12 LP muss gewählt werden. Die angegebene Verteilung der LP auf Lehrveranstaltungen ist nur exemplarisch.

### 3.50 Wahlpflichtmodul: Type Systems for Correctness and Security

Modulname	Type Systems for Correctness and Security / Type Systems for Correctness and Security
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Typsysteme für Korrektheit und Sicherheit : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Typsysteme für Korrektheit und Sicherheit: Vorlesung ( 45h / 105h / EN / WS / 30 ) Typsysteme für Korrektheit und Sicherheit: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Typsysteme für Korrektheit und Sicherheit: Aus der Vorlesung “Grundlagen der Programmiersprachen”: Spracheigenschaften, Syntaktische Strukturen, Datentypen, Funktionale Programmierung	
<b>Inhalte</b>	
<p>Typsysteme für Korrektheit und Sicherheit: Typsysteme in Programmiersprache helfen fehlerhaftes Verhalten von Beginn an zu Vermeiden. Sie erlauben wertvolle Rückmeldungen für Entwickler, um Fehler und Systemabstürze zu vermeiden oder sogar Sicherheitslücken zu erkennen. In dieser Vorlesung werden wir Typsysteme entwickeln und untersuchen. Wir werden die Theorie behandeln, die Eigenschaften mit denen Typsysteme uns in der Softwareentwicklung unterstützen besprechen und die Implementierung aktueller Typsysteme untersuchen.</p> <p>Wir werden einen pragmatischen Ansatz durchführen und im Laufe der Vorlesung und im Rahmen von Übungen die Implementierung von Typcheckern erüben. Weiterhin untersuchen wir die Typsysteme von bekannten Programmiersprachen wie Java oder Scala genauer.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Teilnehmer sind in der Lage die Definition und Implementierung von Typsystemen zu verstehen und selbstständig zu bearbeiten. Die erlernten Kenntnisse werden im Kurs diskutiert und vertieft, so dass die Teilnehmer danach in der Lage sind mit dem erworbenen Faktenwissen und der Methodenkompetenz den Vorlesungsstoff auf andere Problemstellungen anwenden zu können.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>
<b>Methodische Umsetzung</b>
Typsysteme für Korrektheit und Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Diskussionen</li> <li>• Lesen</li> <li>• Übungen mit begleitender Implementierung</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Ben Hermann
<b>Lernmaterialien, Literaturangaben</b>
Typsysteme für Korrektheit und Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Übungsaufgaben</li> </ul> Weiterhin wird folgende Literatur empfohlen: <ul style="list-style-type: none"> <li>• Benjamin C. Pierce. 2002. Types and Programming Languages. The MIT Press.</li> <li>• Benjamin C. Pierce. 2004. Advanced Topics in Types and Programming Languages. The MIT Press.</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.51 Wahlpflichtmodul: Usability Engineering Practice

Modulname	Usability Engineering Practice / Usability Engineering Practice
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Praxis des Usability Engineering : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Praxis des Usability Engineering: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Praxis des Usability Engineering: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Praxis des Usability Engineering: Kenntnisse aus der Vorlesung „Interaktionsdesign“ des Bachelorstudiengangs Informatik werden vorausgesetzt, da hier die grundlegenden Kenntnisse zu Usability-Methoden erarbeitet werden.	
<b>Inhalte</b>	
Praxis des Usability Engineering: Die Vorlesung „Praxis des Usability Engineering“betzt die Bachelorveranstaltung „Interaktionsgestaltung“ fort. Nachdem dort grundlegende Verfahren des Usability Engineering (Benutzertests, Inspektionsmethoden) eingeführt werden, werden nun einzelne ausgewählte und aktuelle Verfahren (z.B. Cognitive Walkthrough, CardSorting, Value-Centered-Design) vertiefend und in praktischen Übungen ausführlich behandelt. Außerdem betrachtet die Veranstaltung Konzepte und Methoden, die den Begriff des Usability Engineering in vielerlei Hinsicht ausgeweitet betrachten, durch Behandlung etwa von User Experience, Extreme Usability, Ästhetik oder Gesundheits- und Sicherheitsaspekten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Zum einen wird konkretes Faktenwissen zu aktuellen Usabilitymethoden und -ansätzen vermittelt. Methodisch erlernen die Hörer die Anwendung dieser Methoden an hinreichend komplexen, realitätsnahen Beispielen. Damit sind die Hörer am Ende in der Lage die Usability von Softwaresystemen systematisch und mit geeigneter Werkzeugunterstützung zu beurteilen, lernen aber auch die Grenzen solcher Beurteilungsvorgänge kennen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	

<b>Methodische Umsetzung</b>
Praxis des Usability Engineering: Die Vorlesung wird klassisch mit Beamerpräsentation gehalten und intensiv durch die E-Learning-Umgebung koaLA der Universität Paderborn unterstützt. Hier werden vor der Vorlesung die Folien veröffentlicht, schriftliche Übungsaufgaben gestellt, Software (Modellierungswerkzeuge) und Videoaufzeichnungen aller Vorlesungen zur Verfügung gestellt. Im Rahmen der Vorlesung selbst findet immer wieder an geeigneter Stelle interaktive Gruppenarbeit statt – etwa zur Ästhetik oder zu Modellkonzepten wie dem Value-Centered-Design-Ansatz. In den Übungsgruppen stellen Hörer die von ihnen erarbeiteten Lösungen vor und zur Diskussion. Dazu müssen die Vortragenden ihre Lösung anhand von geeigneten Vortragsfolien präsentieren.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gerd Szwillus
<b>Lernmaterialien, Literaturangaben</b>
Praxis des Usability Engineering: <ul style="list-style-type: none"> <li>• Vorlesungsfolien als PDF zum Herunterladen</li> <li>• Schriftliche Hausaufgaben</li> <li>• Diverse, während der Vorlesung eingesetzte Modellierungswerkzeuge</li> <li>• Ben Shneiderman: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2009</li> <li>• David Benyon: Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction</li> </ul>



Design, 2009

- Jakob Nielsen und Raluca Budi: Mobile Usability, 2012
- Effie Lai-Chong Law et al. (ed): Maturing Usability: Quality in Software, Interaction and Value, 2007

**Sonstige Hinweise**

keine

### 3.52 Wahlpflichtmodul: Vehicular Networking

Modulname	Vehicular Networking / Vehicular Networking
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Vehicular Networking : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Vehicular Networking: Vorlesung ( 45h / 105h / EN / SS / 40 ) Vehicular Networking: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Vehicular Networking: Systemsoftware und systemnahe Programmierung	
<b>Inhalte</b>	
Vehicular Networking: Dieser Kurs behandelt wichtige Aspekte sowohl von modernen Vernetzungskonzepten im Auto als auch zwischen Fahrzeugen. Dabei werden Aspekte wie Electronic Control Units, Bussysteme, Fahrerassistenzsysteme, aber auch Verkehrsinformationssysteme, Konzepte der drahtlosen Kommunikation bis hin zu Sicherheit und Privatsphäre behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Ziel ist es, grundlegende Konzepte vernetzter eingebetteter Systeme zu verstehen. Die Studierenden sind in der Lage, diese Konzepte anzuwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Vehicular Networking: Vorlesung mit praktischen Übungen	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.	

<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Vehicular Networking: Folien, Lehrbücher, Papiere
<b>Sonstige Hinweise</b>
keine

### 3.53 Wahlpflichtmodul: VLSI Testing

Modulname	VLSI Testing / VLSI Testing
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• VLSI Testing : 1</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
VLSI Testing: Vorlesung ( 30h / 120h / EN / WS / 40 ) VLSI Testing: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
VLSI Testing: Digitaltechnik	
<b>Inhalte</b>	
VLSI Testing: Die Lehrveranstaltung behandelt systematische Verfahren zur Erkennung von Hardware-Defekten in mikroelektronischen Schaltungen. Es werden sowohl Algorithmen zur Erzeugung und Auswertung von Testdaten als auch Hardwarestrukturen zur Verbesserung der Testbarkeit und für den eingebauten Selbsttest vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach Absolvieren des Moduls in der Lage, <ul style="list-style-type: none"> <li>• Fehlermodelle, Maßnahmen zur Verbesserung der Testbarkeit und Werkzeuge zur Unterstützung des Tests zu beschreiben,</li> <li>• die grundlegenden Modelle und Algorithmen für Fehlersimulation und Test zu erklären und anzuwenden, sowie</li> <li>• Systeme im Hinblick auf ihre Testbarkeit zu analysieren und geeignete Teststrategien auszuwählen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
VLSI Testing: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafel</li> <li>• Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen,</li> </ul>	

Präsentation der Lösungen durch Übungsteilnehmer <ul style="list-style-type: none"> <li>• Praktische Übungen mit verschiedenen Software-Werkzeugen am Rechner</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Sybille Hellebrand
<b>Lernmaterialien, Literaturangaben</b>
VLSI Testing: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing for Digital, Memory &amp; Mixed-Signal VLSI Circuits, Boston, Dordrecht, London: Kluwer Academic Publishers, 2000</li> <li>• L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann Series in Systems on Silicon, ISBN: 0123705975</li> <li>• Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien im koala-Kurs</li> </ul>
<b>Sonstige Hinweise</b>
keine



# Anhang A

## Überblickstabellen

### A.1 Studienrichtungen und Module

	Algorithm Design (S. 11)	Computer Systems (S. 12)	Intelligence and Data (S. 13)	Networks and Communication (S. 14)	Software Engineering (S. 15)
Adaptive Hardware and Systems (S. 18)	-	X	-	-	-
Advanced Algorithms (S. 21)	X	-	-	-	-
Advanced Compiler Construction (S. 23)	-	-	-	-	X
Advanced Complexity Theory (S. 25)	X	-	-	-	-
Advanced Computer Architecture (S. 28)	-	X	-	-	-
Advanced Distributed Algorithms and Data Structures (S. 30)	X	-	-	X	-
Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 32)	-	-	-	-	X
Algorithmic Game Theory (S. 34)	X	-	-	-	-
Algorithms for Highly Complex Virtual Scenes (S. 36)	X	-	-	-	-
Algorithms for Synthesis and Optimization of Integrated Circuits (S. 39)	-	X	-	-	-
Architektur paralleler Rechnersysteme (S. 41)	-	X	-	-	-
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 44)	-	-	-	X	-
Build It, Break It, Fix It (S. 47)	-	-	-	-	X
Clustering Algorithms (S. 50)	X	-	X	-	-
Compiler Construction (S. 52)	-	-	-	-	X
Computational Geometry (S. 54)	X	-	-	-	-

Designing code analyses for large-scale software systems (S. 58)	-	-	-	-	X
Empiric performance evaluation (S. 61)	-	X	-	X	X
Foundations of Cryptography (S. 63)	X	-	-	-	-
Fundamentals of Model-Driven Engineering (S. 65)	-	-	-	-	X
Future Internet (S. 68)	-	-	-	X	-
Hardware/Software Codesign (S. 70)	-	X	-	-	-
High-Performance Computing (S. 72)	-	X	-	-	X
Intelligence in Embedded Systems (S. 74)	-	X	X	-	-
Interactive Data Visualization (S. 76)	-	-	X	-	-
Kontextuelle Informatik (S. 79)	-	-	-	-	X
Language-Based Security (S. 82)	-	-	-	-	X
Linear and Integer Optimization (S. 84)	X	-	-	-	-
Logic and Automated Reasoning (S. 86)	-	-	X	-	-
Logic Programming for Artificial Intelligence (S. 89)	-	-	X	-	X
Machine Learning I (S. 92)	-	-	X	-	-
Machine Learning II (S. 94)	-	-	X	-	-
Master-Abschlussarbeit (S. 96)	-	-	-	-	-
Mobile Communication (S. 99)	-	-	-	X	-
Model Checking (S. 101)	-	-	-	-	X
Model-Based Interface Development (S. 103)	-	-	-	-	X
Network Simulation (S. 106)	-	-	-	X	-
Networked Embedded Systems (S. 108)	-	-	-	X	-
Planning and Heuristic Search (S. 110)	-	-	X	-	-
Projektgruppe (S. 113)	-	-	-	-	-
Public-Key Cryptography (S. 116)	X	-	-	-	-
Reconfigurable Computing (S. 118)	-	X	-	-	-
Routing and Data Management in Networks (S. 120)	X	-	-	X	-
Seminar I (S. 122)	-	-	-	-	-
Seminar II (S. 124)	-	-	-	-	-
Software Analysis (S. 126)	-	-	-	-	X
Software Quality Assurance (S. 128)	-	-	-	-	X
Studium Generale (S. 130)	-	-	-	-	-
Type Systems for Correctness and Security (S. 132)	-	-	-	-	X



Usability Engineering Practice (S. 135)	-	-	-	-	X
Vehicular Networking (S. 138)	-	-	-	X	-
VLSI Testing (S. 140)	-	X	-	-	-

## A.2 Module und Lehrveranstaltungen

Adaptive Hardware and Systems (S. 18)		Adaptive Hardware and Systems (S. 77)	
Advanced Algorithms (S. 21)		Advanced Algorithms (S. 77)	
Advanced Compiler Construction (S. 23)		Advanced Compiler Construction (S. 77)	
Advanced Complexity Theory (S. 23)		Advanced Complexity Theory (S. 77)	
Advanced Computer Architecture (S. 28)		Advanced Computer Architecture (S. 77)	
Advanced Distributed Algorithms and Data Structures (S. 30)		Advanced Distributed Algorithms and Data Structures (S. 77)	
Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 32)		Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 77)	
Algorithmic Game Theory (S. 34)		Algorithmic Game Theory (S. 77)	
Algorithms for High Complex Virtual Scenes (S. 36)		Algorithmen für hochkomplexe virtuelle Szenen (S. 77)	
Algorithms for Synthesis and Optimization of Integrated Circuits (S. 39)		Algorithms for Synthesis and Optimization of Integrated Circuits (S. 77)	
Architektur paralleler Rechensysteme (S. 41)		Architektur paralleler Rechensysteme (S. 77)	
Bitcoin, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 44)		Bitcoin, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 77)	
Build It, Break It, Fix It (S. 47)		Build It, Break It, Fix It (S. 77)	
Clustering Algorithms (S. 50)		Clustering Algorithms (S. 77)	
Compiler Construction (S. 52)		Compilerbau (S. 77)	
Computational Geometry (S. 54)		Algorithmische Geometrie (S. 77)	
Deductive Verification (S. 56)		Deductive Verification (S. 77)	
Designing code analyses for large-scale software systems (S. 59)		Designing code analyses for large-scale software systems (S. 77)	
Empirical performance evaluation (S. 61)		Empirische Leistungsbewertung (S. 77)	
Foundations of Cryptography (S. 63)		Foundations of Cryptography (S. 77)	
Fundamentals of Model-Driven Engineering (S. 65)		Fundamentals of Model-Driven Engineering (S. 77)	
Future Internet (S. 68)		Future Internet (S. 77)	
Hardware/Software Codesign (S. 70)		Hardware/Software Codesign (S. 77)	
High-Performance Computing (S. 72)		High-Performance Computing (S. 77)	
Intelligence in Embedded Systems (S. 74)		Intelligenz in eingebetteten Systemen (S. 77)	
Interactive Data Visualization (S. 76)		Interaktive Datenvisualisierung (S. 77)	
Kontextuelle Informatik (S. 79)		Kontextuelle Informatik (S. 77)	
Language-Based Security (S. 82)		Language-Based Security (S. 77)	
Linear and Integer Optimization (S. 84)		Linear and Integer Optimization (S. 77)	
Logic- and Automated Reasoning (S. 86)		Logic Programming for Artificial Intelligence (S. 77)	
Logic Programming for Artificial Intelligence (S. 88)		Logik und automatisches Beweisen (S. 77)	
Machine Learning I (S. 90)		Maschinelles Lernen I (S. 77)	
Machine Learning II (S. 91)		Maschinelles Lernen II (S. 77)	
Master-Abschleissarbeit (S. 96)		Masterarbeit (S. 77)	
Mobile Communication (S. 99)		Mobile Kommunikation (S. 77)	
Model Checking (S. 101)		Model Checking (S. 77)	
Model-Based Interface Development (S. 103)		Modellbasierte Entwicklung von Benutzungsschnittstellen (S. 77)	
Network Simulation (S. 106)		Network Simulation (S. 77)	
Networked Embedded Systems (S. 108)		Networked Embedded Systems (S. 77)	
Planning and Heuristic Search (S. 110)		Planen und Heuristische Suche (S. 77)	
Projektgruppe (S. 113)		Projektgruppe (S. 77)	
Public-Key Cryptography (S. 116)		Public-Key Cryptography (S. 77)	
Reconfigurable Computing (S. 118)		Reconfigurable Computing (S. 77)	
Routing and Data Management in Networks (S. 120)		Routing and Data Management in Networks (S. 77)	
Seminar I (S. 122)		Seminar (S. 77)	
Seminar II (S. 124)		Seminar II (S. 77)	
Software Analysis (S. 126)		Software Analysis (S. 77)	
Software Quality Assurance (S. 128)		Software Quality Assurance (S. 77)	
Studium Generale (S. 130)		Studium Generale (S. 77)	
Type Systems for Correctness and Security (S. 132)		Type Systems for Correctness and Security (S. 77)	
Usability Engineering Practice (S. 135)		Praxis des Usability Engineering (S. 77)	
Vehicle Networking (S. 138)		Vehicle Networking (S. 77)	
VLSI Testing (S. 140)		VLSI Testing (S. 77)	