

UNIVERSITÄT PADERBORN

FAKULTÄT FÜR ELEKTROTECHNIK, INFORMATIK UND MATHEMATIK
INSTITUT FÜR INFORMATIK

MODULHANDBUCH FÜR DEN
BACHELORSTUDIENGANG INFORMATIK V4 (IBA v4)

STAND: 11. SEPTEMBER 2023

Inhaltsverzeichnis

1	Präambel und Hinweise	3
2	Erster Studienabschnitt	4
2.1	Softwaretechnik	4
2.2	Algorithmen und Komplexität	20
2.3	Computersysteme	28
2.4	Mathematik	38
3	Zweiter Studienabschnitt	48
3.1	Algorithmen und Komplexität	48
3.2	Computersysteme	61
3.3	Daten und Wissen	69
3.4	Softwaretechnik	80
4	Weiteres	91
5	Module im Wintersemester	98
6	Module im Sommersemester	99
7	Modules in English	100

1 Präambel und Hinweise

Aus technischen Gründen wurde die Präambel des Modulhandbuches ausgelagert. Sie ist unter Modulhandbuch Informatik auf den Seiten zum Studium des Instituts für Informatik zu finden. Wir bitten um Beachtung dieser Präambel. Bei Fragen zu dieser Präambel wenden Sie sich bitte an die Fachberatung Informatik.

Bitte beachten Sie auch, dass

1. in diesem Modulhandbuch alle laut Prüfungsordnung vorgesehenen Module aufgelistet werden, auch wenn sie in dem entsprechenden Semester nicht angeboten werden.
2. dieses Modulhandbuch den Datenbestand zum Zeitpunkt der Erstellung beinhaltet. Alle Angaben sind ohne Gewähr.

2 Erster Studienabschnitt

Im ersten Studienabschnitt werden die wesentlichen Grundkenntnisse und Fähigkeiten vermittelt, die jeder Absolvent der Informatik kennen und beherrschen sollte. Der erste Studienabschnitt besteht nur aus Pflichtveranstaltungen, die mit einer Ausnahme in den ersten vier Fachsemestern liegen.

2.1 Softwaretechnik

Programmierung							
Programming							
Modulnummer: M.079.1110	Workload (h): 240	Leistungspunkte: 8	Turnus: Wintersemester				
	Studiensemester: 1. Semester	Dauer (in Sem.): 1	Sprache: de				
1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.079.05100 Programmierung	V4 Ü2	90	150	P	500/40	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: keine						

2 Erster Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Programmierung:</i> Softwareentwicklung ist ein zentrales Arbeitsgebiet der Informatik. Software-Entwickler müssen Aufgaben analysieren und modellieren, Software-Strukturen entwerfen und diese in einer Programmiersprache implementieren können. Dieser Modul vermittelt einführende und wissenschaftlich fundierte Kenntnisse und Fähigkeiten in der Programmierung. Zusammen mit den Modulen Programmierung 2, Programmiersprachen, Softwareengineering und dem Softwaretechnikpraktikum werden damit die wissenschaftlichen Grundlagen für das Arbeitsgebiet Software-Entwicklung gelegt und praktisch eingeübt. Dieses Modul soll die Teilnehmer befähigen,</p> <ul style="list-style-type: none">• eine für die Software-Entwicklung relevante Programmiersprache anzuwenden (zur Zeit Python, in geringerem Umfang auch Java)• Grundbegriffe der objektorientierten Programmiermethodik einzusetzen,• Algorithmen in Programmen zu implementieren. <p>Im Informatikstudium bildet dieses Modul zusammen mit den Pflichtmodulen Modellierung, Datenbanksysteme und Softwaretechnik den Kern der Grundausbildung in Gebiet Softwaretechnik. Dieses Modul umfasst die folgenden Inhalte:</p> <ol style="list-style-type: none">1. Grundbegriffe zu Programmen und ihrer Ausführung2. Klassen, Objekte, Datentypen3. Programm- und Datenstrukturen4. Objektorientierte Abstraktion5. Objektorientierte Bibliotheken
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden lernen</p> <ul style="list-style-type: none">• <i>Faktenwissen:</i> unter anderem die wesentlichen Konstrukte einer Programmiersprache (derzeit Python, in geringem Umfang auch Java); die Grundkonzepte von Komposition und Abstraktion in der Programmierung zu verstehen• <i>methodisches Wissen:</i> die gelernten Sprachkonstrukte sinnvoll und mit Verständnis anzuwenden; Software zu testen sowie Fehlerursachen zu finden und zu beseitigen; objektorientierte Grundkonzepte zu verstehen und anzuwenden; Software aus objektorientierten Bibliotheken wiederzuverwenden• <i>Transferkompetenz:</i> praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben zu übertragen• <i>normativ-bewertenden Kompetenzen:</i> den Aufwand und die Durchführbarkeit von Programmieraufgabe zu beurteilen <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Gruppenarbeit• Kooperationskompetenz• Lernmotivation

2 Erster Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur	120-180 min	100%
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.			
7	Studienleistung, qualifizierte Teilnahme:		
zu	Form	Dauer bzw. Umfang	SL / QT
a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Prof. Dr. Stefan Böttcher		
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Programmierung:</i> Methodische Umsetzung Sprachkonstrukte und Programmiertechniken werden an typischen Beispielen eingeführt und erläutert und anschließend in den Übungen praktisch erprobt. Objektorientierte Methoden und Abstraktion werden überwiegend an der Benutzung von Bibliotheken erklärt. In Übungsstunden in Kleingruppen werden praktische Programmieraufgaben unter Anleitung an Rechnern bearbeitet.		

Programmiersprachen			
Programming Languages			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.1111	120	4	Wintersemester

2 Erster Studienabschnitt

	Studiensemester: 1. Semester	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05101 Programmiersprachen	V2 Ü1	45	75	P	500/40
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Programmiersprachen:</i> Empfohlene Vorkenntnisse Die Veranstaltung Programmierung bzw. Kenntnisse im Umgang mit (mindestens einer) Programmiersprache.					
4	Inhalte: <i>Inhalte der Lehrveranstaltung Programmiersprachen:</i> In der Veranstaltung "Programmierung" haben Sie bereits eine grundlegende Einführung in die Programmierung mit <i>einer</i> sogenannten <i>imperativen</i> Programmiersprache genossen, die Berechnungen durch Sequenzen einzelner Befehle beschreiben. Die Veranstaltung "Programmiersprachen" baut hierauf auf und vermittelt Einblicke in <i>andere</i> wichtige sogenannte Programmierparadigmen, speziell in die funktionale Programmierung und Logikprogrammierung. Inhaltsübersicht: <ul style="list-style-type: none"> • Grundlegende Konzepte der Funktionalen Programmierung • Seiteneffekte, Programmieren ohne Seiteneffekte • Rekursion und Aggregation • Grundlegende Konzepte der Logikprogrammierung • Auswertung von Ausdrücken und Regeln, Unifizierung • Logikprogrammierung in Prolog • Eingeschränkte aber effiziente Auswertung in Datalog 					

5	<p>Lernergebnisse und Kompetenzen:</p> <ul style="list-style-type: none"> • Die Studierenden verstehen die grundlegenden syntaktischen und semantischen Konzepte von Programmiersprachen. • Sie können Programme und die darin verwendeten Sprachkonstrukte in angemessen präziser Weise erläutern. • Sie sind in der Lage, einfache Grammatiken, Typspezifikationen und funktionale Programme zu entwickeln. • Sie besitzen die Fähigkeit neue Programmier- und Anwendungssprachen selbstständig zu erlernen. • Sie haben praktisch mit einer Reihe verschiedener Programmiersprachen gearbeitet. • Sie haben Einblick gewonnen in die Probleme des Entwurfs neuer Programmier- und Anwendungssprachen. • Sie können typische Eigenschaften von objektorientierten und funktionalen Sprachen erklären. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernmotivation • Einsatz und Engagement • Gruppenarbeit • Kooperationskompetenz • Schreib- und Lesekompetenz (wissenschaftlich) • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" data-bbox="277 1196 1422 1341"> <thead> <tr> <th>zu</th> <th>Prüfungsform</th> <th>Dauer bzw. Umfang</th> <th>Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Klausur</td> <td>60-90 min</td> <td>100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	60-90 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	60-90 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" data-bbox="277 1498 1422 1644"> <thead> <tr> <th>zu</th> <th>Form</th> <th>Dauer bzw. Umfang</th> <th>SL / QT</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td>SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								

2 Erster Studienabschnitt

10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Prof. Dr. Eric Bodden, Prof. Dr. Stefan Böttcher
13	Sonstige Hinweise: keine

Datenbanksysteme						
Database Systems						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.1114	150	5	Sommersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	2. Semester	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05203 Datenbanksysteme	V2 Ü2	60	90	P	400/40
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	<i>Teilnahmevoraussetzungen der Lehrveranstaltung Datenbanksysteme:</i> Empfohlene Vorkenntnisse Kenntnisse in der Programmierung werden in dem Umfang vorausgesetzt, wie sie in den Veranstaltungen Programmierung und Programmiersprachen gelehrt werden. Elementare Kenntnisse der Logik der Modellierung aus der Vorlesung Modellierung werden ebenfalls vorausgesetzt.					

2 Erster Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Datenbanksysteme:</i></p> <p>Datenbanken spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissen der Unternehmen als Daten in Datenbanken gespeichert wird. Für das Unternehmen ist es entscheidend, dass diese Daten korrekt, insbesondere konsistent, sind und dass sie effizient erfragt und aktualisiert werden können. Weiterhin sind die in Datenbanken abgelegten Datenbestände die wesentliche Datenquelle für eine Vielzahl von Anwendungsprogrammen, sie werden aber auch durch Anwendungsprogramme aktualisiert. Deshalb kommt der Organisation und Verarbeitung großer Datenbestände sowie der Einbindung von Datenbanken in Anwendungen eine zentrale Rolle bei der Erstellung korrekter und effizienter Anwendungen zu. Dieses Modul erschließt die Grundlagen für Datenbanksysteme, die in nahezu allen Unternehmen in der Praxis eingesetzt werden.</p> <ul style="list-style-type: none">• Relationales Datenmodell, relationale Algebra und relationale Kalküle• SQL (Datendefinitionssprache, Datenmanipulationssprache und Anfragesprache)• Eingebettetes SQL• Sichten, Zugriffskontrolle und View-Update-Problematik• Anfrageoptimierung• Datenintegrität• Funktionale Abhängigkeiten und Datenbankschemaentwurf• Transaktionen (Synchronisation und Recovery)• NoSQL-Datenbanken
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Studierende lernen</p> <ul style="list-style-type: none">• <i>Faktenwissen:</i> Theorie und Konzepte relationaler Anfragesprachen kennen; Konzepte des Datenbankentwurfs kennen; Konzepte der Synchronisation und Recovery von Transaktionen kennen• <i>Vermittlung von methodischem Wissen in Kleingruppen-Präsenz-Übungen:</i> komplexe Anfragen an relationale Datenbanken korrekt formulieren; ein Datenbankschema möglichst redundanzfrei entwerfen• <i>in praktischen Übungen am Rechner:</i> eigene SQL-Anfragen an existierende relationale Datenbanken stellen; Programme zu schreiben, die Datenbestände aus Datenbanken lesen oder verändern; eigene Datenbanken zu definieren und aufzubauen• <i>Vermittlung von Transferkompetenz:</i> die erworbenen Kompetenzen und Fertigkeiten auf andere Datenquellen oder andere Datenbanksysteme übertragen; Umgang mit Zugriffsrechten• <i>Vermittlung von normativ-bewertenden Kompetenzen:</i> die Eignung und Grenzen des relationalen Datenmodells bewerten und einschätzen; den Programmieraufwand für Datenbankabfragen und Datenbankprogrammierung einschätzen; die Folgen einer Datenbankschema-Änderung erkennen und abschätzen; die Risiken eines schlecht entworfenen Datenbankschemas bewerten; den Aufwand und Nutzen von Synchronisation und Recovery abschätzen <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Gruppenarbeit• Lernkompetenz• Lernmotivation

2 Erster Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur	60-90 min	100%
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.			
7	Studienleistung, qualifizierte Teilnahme:		
zu	Form	Dauer bzw. Umfang	SL / QT
a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Dr. Rita Hartel, Prof. Dr. Stefan Böttcher		
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Datenbanksysteme:</i> Methodische Umsetzung Die Grundlagen und Konzepte von Datenbanksystemen werden im Rahmen einer Vorlesung eingeführt und anschließend in Präsenzübungen in Kleingruppen sowie in Heimübungen vertieft und durch praktische Übungen ergänzt. Lernmaterialien, Literaturangaben <ul style="list-style-type: none"> • Lehrbuch: Kemper, Eickler: Datenbanksysteme , Oldenbourg-Verlag, neueste Ausgabe. • Lehrbuch: Garcia-Molina, Ullman, Widom: Database Systems: The Complete Book, Prentice Hall, neueste Ausgabe. 		

Software Engineering

Software Engineering

2 Erster Studienabschnitt

Modulnummer: M.079.1112	Workload (h): 150	Leistungspunkte: 5	Turnus: Sommersemester			
	Studiensemester: 2. Semester	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05202 Software Engineering	V2 Ü1	45	75	P	300/30
b)	L.079.05206 Praktikum: Software Engineering	P	0	30	P	3
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Software Engineering:</i> Empfohlene Vorkenntnisse Programmierung, Modellierung <i>Teilnahmevoraussetzungen der Lehrveranstaltung Praktikum: Software Engineering:</i> Empfohlene Vorkenntnisse Programmierung, Modellierung					

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Software Engineering:</i></p> <p>In der Vorlesung werden die Grundlagen der systematischen und ingenieurmäßigen Softwareentwicklung vermittelt. Im Fokus steht dabei die modellbasierte Softwareentwicklung. Die Vorlesung führt in wesentliche Vorgehensmodelle für die Softwareentwicklung ein, sowohl klassische als auch agile. Es werden Methoden für die Softwareentwicklung und -qualitätssicherung vermittelt, die innerhalb der Vorgehensmodelle zum Einsatz kommen. Außerdem werden Modellierungssprachen und Softwarewerkzeuge vorgestellt, mit denen die statischen und dynamischen Aspekte von Softwaresystemen beschrieben werden können. Insbesondere wird die objektorientierte Modellierungssprache UML (Unified Modeling Language) eingeführt, die unterschiedliche Diagrammsprachen wie Klassendiagramme, Komponentendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme und Zustandsdiagramme vereint. Modellierungswerkzeuge werden exemplarisch eingesetzt.</p> <p>Die Vorlesung wird abgerundet durch eine durchgängige Entwicklungsmethode von der Anforderungsspezifikation über den Architektur- und Softwareentwurf bis hin zur Implementierung und dem Testen der Software. Hierbei wird vor allem auf die Aspekte der systematischen Ableitung und Verfeinerung von Modellen, der Transformation von Modellen in Programmcode (Codegenerierung) sowie des modellbasierten Testens eingegangen. Es werden methodische Hinweise zur Erstellung der Ergebnisartefakte (u.a. Richtlinien, Architekturstile und Entwurfsmuster) und zur Prüfung ihrer Qualität sowie zum Einsatz der Modellierungssprachen im Softwareentwicklungsprozess gegeben. Darüber hinaus werden Techniken zur Definition und domänenspezifischen Anpassung von Modellierungssprachen (Metamodellierung, UML-Profile sowie Beispiele konkreter domänenspezifischer Sprachen (DSLs) wie SysML oder BPMN) betrachtet.</p> <p>Die Vorlesung wird durch Übungen begleitet, in denen die Vorlesungsinhalte aufgegriffen, vertieft und an beispielhaften Entwicklungsaufgaben selbst angewendet werden. Schwerpunkte der Veranstaltung sind:</p> <ul style="list-style-type: none">• Vorgehensmodelle (klassische, agile)• UML (Unified Modelling Language): Klassendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme, Zustandsdiagramme• Modellbasiertes Vorgehensmodell• durchgängige Softwareentwicklungsmethode von der Anforderungsspezifikation über Modellierung bis zur Implementierung und dem Test der Software• Modellbasiertes Testen• Domänenspezifische Sprachen (Metamodellierung, UML Profile, SysML, BPMN) <p><i>Inhalte der Lehrveranstaltung Praktikum: Software Engineering:</i></p> <p>Begleitend zur Vorlesung Software Engineering vertiefen die Studierenden anhand von praktischen Übungsaufgaben ihre Kenntnisse in der Programmiersprache Java, welche häufig zur systematischen Entwicklung größerer Softwaresysteme verwendet wird. Ein Schwerpunkt liegt dabei auf der technischen Umsetzung von Objektorientierten Konzepten und UML-Spezifikationen (insbesondere Klassendiagrammen) in ausführbare Softwareapplikationen.</p> <p>Die Studierenden erlernen anhand praktischer Beispiele die Zusammenhänge zwischen Entwurf und Implementierung von Softwareanwendungen. Außerdem erwerben sie die Kompetenz, bekannte Programmierkonzepte, welche anhand der Sprache Python erlernt wurden, auf eine weitere Programmiersprache und neue Anwendungsgebiete zu transferieren.</p>
---	---

2 Erster Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sollen in der Lage sein, für ein gegebenes Problem schrittweise eine Softwarelösung zu entwickeln. Hierzu sollen sie ein modellbasiertes Vorgehen einsetzen können, wobei sie für die einzelnen Entwicklungsschritte unterschiedliche Diagrammart der UML (Unified Modeling Language) verwenden. Zur Überprüfung der Qualität der entwickelten Softwarelösung sollen sie in der Lage sein, Techniken des modellbasierten Testens einzusetzen. Sie verstehen die Beziehungen und Übergänge zwischen verschiedenen Entwicklungsphasen eines Vorgehensmodells. Sie beherrschen verschiedene Diagrammsprachen der UML zur Modellierung der unterschiedlichen Aspekte einer Softwarelösung und können die Qualität von Zwischenergebnissen bewerten. Außerdem haben sie ein grundlegendes Verständnis der Techniken zur Entwicklung und Spezialisierung von Modellierungssprachen für spezielle Situationen und Domänen.</p> <p>Durch den Einsatz des Gelernten am durchgängigen Beispiel der Praktikumsaufgabe verstehen die Studierenden die Bedeutung der verschiedenen Phasen einer Softwareentwicklung und sind in der Lage, diese durchgängig an einem konkreten Softwaresystem einzusetzen.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Einsatz und Engagement • Haltung und Einstellung • Kooperationskompetenz • Lernkompetenz • Lernmotivation • Selbststeuerungskompetenz 												
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td>a) - b)</td> <td>Klausur</td> <td>60-90 min</td> <td>100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a) - b)	Klausur	60-90 min	100%				
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote										
a) - b)	Klausur	60-90 min	100%										
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td>SL</td> </tr> <tr> <td>b)</td> <td>Praktikumsarbeit mit anschließendem Gespräch</td> <td></td> <td>QT</td> </tr> </tbody> </table> <p>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>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL	b)	Praktikumsarbeit mit anschließendem Gespräch		QT
zu	Form	Dauer bzw. Umfang	SL / QT										
a)	Schriftliche Übungsaufgaben		SL										
b)	Praktikumsarbeit mit anschließendem Gespräch		QT										
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>												
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>												

2 Erster Studienabschnitt

10	<p>Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</p>
11	<p>Verwendung des Moduls in anderen Studiengängen: keine</p>
12	<p>Modulbeauftragte/r: Dr. Stefan Sauer</p>
13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Software Engineering:</i> Methodische Umsetzung In der Vorlesung werden die Grundlagen, Begrifflichkeiten, Sprachen und Methoden des Software Engineering vermittelt, die dann in den begleitenden Übungen vertieft und in dem begleitenden Praktikumsanteil von den Studierenden an einem durchgängigen Beispiel selbst erprobt werden. Lernmaterialien, Literaturangaben Folien, Tafelanschrieb, evtl. Vorlesungsaufzeichnung, Übungen, Praktikumsaufgabe (siehe Praktikum: Software Engineering)</p> <p><i>Hinweise der Lehrveranstaltung Praktikum: Software Engineering:</i> Methodische Umsetzung Softwareentwicklungsaufgaben (Modellieren, Implementieren, Testen) in kleinen Teams. Lernmaterialien, Literaturangaben Aufgabenstellung, Material der begleitenden Vorlesung Software Engineering, eigenständige Recherche zu weiterführender Literatur (in der Bibliothek, im Internet)</p>

Softwaretechnikpraktikum							
Software Engineering Project							
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:				
M.079.1113	240	8	Wintersemester				
Studiensemester:		Dauer (in Sem.):	Sprache:				
3. Semester		1	de				
1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
a)	L.079.05300 Softwaretechnikpraktikum	V1 Ü2	45	195	P	150/10	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						

2 Erster Studienabschnitt

3	<p>Teilnahmevoraussetzungen:</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Softwaretechnikpraktikum:</i> Die Module Programmierung und Software Engineering müssen bestanden sein.</p> <p>Empfohlene Vorkenntnisse</p> <ul style="list-style-type: none">• Objektorientierte Programmierung mit Java• Modellbasierter Softwareentwurf mit UML
4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Softwaretechnikpraktikum:</i> Das Softwaretechnikpraktikum ist eine praxisorientierte Lehrveranstaltung inklusive Vorlesungen zum Thema Projektmanagement. Eine komplexe Softwareentwicklungsaufgabe wird im Team von ca. zehn Studierenden unter Verwendung von UML und Java bearbeitet. Schwerpunkt des Praktikums ist die Vermittlung von Erfahrungen mit der gruppenbasierten Softwareentwicklung unter Benutzung marktüblicher Werkzeuge, Methoden und Prozesse.</p> <ul style="list-style-type: none">• Planung des Projektmanagements• Erstellung von Projektplänen, Aufwandsschätzungen, Anforderungsspezifikationen und Entwurfsdokumentationen• Umsetzung der Softwareentwicklungsaufgabe unter Einsatz etablierter Softwareentwicklungsmethoden, wie z.B. Scrum• Protokollierung des Aufwandes durch Stundenzettel
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sollen Sprachen und Werkzeuge im Softwareentwicklungsprozess einsetzen sowie den organisatorischen Ablauf eines Softwareprojekts von der Anforderungsdefinition bis zur Abgabe kennen lernen. Darüber hinaus sollen die Studierenden den praktischen Nutzen von planerisch durchdachten Projekten, sowie die Probleme gruppenorientierter Softwareentwicklung und erste Ansätze zu ihrer Bewältigung kennen lernen.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Einsatz und Engagement• Gruppenarbeit• Kooperationskompetenz• Selbststeuerungskompetenz

2 Erster Studienabschnitt

6	Prüfungsleistung: <input type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input checked="" type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur, Softwareprojekte	60-90 min	
<p>Im Softwaretechnikpraktikum 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.</p> <p>Die Prüfungsleistung besteht aus einer Klausur im Umfang von 60-90 Minuten, die 30% der Modulnote bildet, sowie Softwareprojekten mit Dokumentation, die 70% der Modulnote bilden. Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>			
7	Studienleistung, qualifizierte Teilnahme:		
zu	Form	Dauer bzw. Umfang	SL / QT
a)	Praktikumsarbeit		SL
<p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der Hälfte der Anzahl seiner Credits gewichtet (Faktor 0,5).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Prof. Dr. Eric Bodden		

2 Erster Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Softwaretechnikpraktikum:</i></p> <p>Methodische Umsetzung Durchführung eines Projekts mit regelmäßiger Abgabe von Arbeitsergebnissen (Modulteilprüfungen), protokollierten Gruppensitzungen und einer Abschlusspräsentation.</p> <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Balzert, H. (2008). Lehrbuch der Softwaretechnik. Bd. 3: Softwaremanagement. Hrsg. von C. Ebert. 2. Aufl. Lehrbücher der Informatik. Heidelberg: Spektrum. ISBN: 978-3-8274-1161-7. • Balzert, H. (2011). Lehrbuch der Softwaretechnik. Bd. 2: Entwurf, Implementierung, Installation und Betrieb. 3. Aufl. Lehrbücher der Informatik. Heidelberg: Spektrum. ISBN: 978-3-8274-1706-0. • Sommerville, I. (2012). Software Engineering. 9. Aufl. Always Learning. München: Pearson. ISBN: 978-3-86894-099-2.
----	---

Gestaltung von Nutzungsschnittstellen						
Designing User Interfaces						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.1115	180	6	Sommersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	4. Semester	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05400 Gestaltung von Nutzungsschnittstellen	V4 Z1	75	105	P	300
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	keine					

2 Erster Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Gestaltung von Nutzungsschnittstellen:</i> Erlernbarkeit, Beeinträchtigungsfreiheit sowie die barrierefreie Erschließbarkeit von Softwaresystemen sind heute Pflichtanforderungen an die Systemgestaltung. Grundlegende Herausforderungen sind, mit Hilfe geeigneter Gestaltungsmaßnahmen Verständnisprozesse bei Nutzern zu fördern und unnötige Belastungen bei der Arbeit mit Softwaresystemen zu vermeiden. Dazu ist ein methodisches Repertoire erforderlich, um schon während des Entwurfs die Gebrauchstauglichkeit sichern zu können. Die dazu erforderlichen Kenntnisse und Fertigkeiten reichen von physiologischen und psychologischen Grundlagen über Methoden und Techniken der Systemgestaltung bis hin zu rechtlichen Anforderungen.</p> <ul style="list-style-type: none"> • Gesetzliche Regelungen und Normen: EU Bildschirmrichtlinie (90/270/EWG), Bildschirmarbeitsverordnung (BildscharbV), Barrierefreie Informations-Technik Verordnung (BITV), DIN EN ISO 9241 etc. • Kognitionspsychologische Grundlagen: Wahrnehmung, Aufmerksamkeit, Gedächtnis etc. • Physiologische Grundlagen: Sensorik, Motorik • Konzepte: Interaktionstechniken, Farbmodelle, verteilte und natürliche Benutzungsoberflächen • Gestaltungsempfehlungen: Normen (z. B. DIN EN ISO 9241) Leitlinien, Kriterien • Usability Engineering: Konstruktionsmethoden, Analysemethoden, Web Usability 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sind in der Lage, grundlegende Problembereiche der Mensch-Rechner-Interaktion zu erkennen und sie konstruktiv gestaltend umzusetzen. Sie erwerben zugleich anschlussfähiges Wissen, das vor allem für die Zusammenarbeit mit Designern und Psychologen erforderlich ist, aber auch für den Diskurs mit Medienwissenschaftlern und Pädagogen (E-Learning) hilfreich ist. Die erworbenen Kenntnisse und Fertigkeiten bilden zugleich die Grundlage für vertiefende Veranstaltungen im Bereich der Mensch-Maschine-Wechselwirkung wie z. B. Usability Engineering, Computergrafik oder auch Medien-Ergonomie.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Empathie • Haltung und Einstellung • Medienkompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">90-120 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	90-120 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	90-120 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <p>keine</p>								

2 Erster Studienabschnitt

8	Voraussetzungen für die Teilnahme an Prüfungen: keine
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Dr. Felix Winkelkemper
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Gestaltung von Nutzungsschnittstellen:</i> Methodische Umsetzung: Die Veranstaltung wird entsprechend dem Flipped-Classroom-Konzept angeboten. Die Vorlesungsinhalte werden durch Videos vermittelt, die flexibel angeschaut werden können. Eine zentrale Präsenzveranstaltung dient der Klärung von Fragen und der Vertiefung des Lehrinhalts in kleinen Aufgaben. Übungsblätter werden von Tutoren bewertet und dienen der individuellen Rückmeldung. Lernmaterialien, Literaturangaben: Vorlesungsvideos, Vorlesungsfolien und Übungsblätter. Weitere Zusatzmaterialien werden in der Veranstaltung bekanntgegeben.

2.2 Algorithmen und Komplexität

Modellierung						
Modelling						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.1210	240	8	Wintersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	1. Semester	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05102 Modellierung	V4 Ü2	90	150	P	500/40

2 Erster Studienabschnitt

2	<p>Wahlmöglichkeiten innerhalb des Moduls:</p> <p>keine</p>								
3	<p>Teilnahmevoraussetzungen:</p> <p>keine</p>								
4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Modellierung:</i></p> <p>Das Modellieren ist eine für das Fach Informatik typische Arbeitsmethode, die in allen Gebieten des Faches angewandt wird. Aufgaben, Probleme oder Strukturen werden untersucht und als Ganzes oder in Teilaspekten beschrieben, bevor sie durch den Entwurf von Software, Algorithmen, Daten und/oder Hardware gelöst bzw. implementiert werden. Mit der Modellierung eines Problems zeigt man, ob und wie es verstanden wurde. Damit ist sie Voraussetzung und Maßstab für die Lösung und sie liefert meist auch den Schlüssel für einen systematischen Entwurf. Als Ausdrucksmittel für die Modellierung steht ein breites Spektrum von Kalkülen und Notationen zur Verfügung. Sie sind spezifisch für unterschiedliche Arten von Problemen und Aufgaben. Deshalb werden in den verschiedenen Gebieten der Informatik unterschiedliche Modellierungsmethoden eingesetzt. In den entwurfsorientierten Gebieten (Softwaretechnik, Hardware-Entwurf) ist die Bedeutung der Modellierung und die Vielfalt der Methoden natürlich besonders stark ausgeprägt.</p> <ul style="list-style-type: none"> • Grundlegende Kalküle: Wertebereiche, Terme, Algebren • Logik: Aussagenlogik, Prädikatenlogik • Modellierung mit Graphen: Weg, Verbindung, Zuordnung, Abhängigkeiten, Abfolgen • Grammatiken: reguläre und kontextfreie Grammatiken • Modellierung von Abläufen: endliche Automaten, Petri-Netze • Modellierung von Unsicherheit 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Studierende kennen wesentliche Techniken zur Modellierung informatischer Probleme. Sie können für ein gegebenes Problem eine geeignete Modellierungstechnik auswählen und das Problem mit dieser Technik beschreiben. Sie können grundlegende Techniken erweitern und verfeinern, um so neuartige Probleme zu modellieren.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernkompetenz • Motivationale und volitionale Fähigkeiten 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">120-180 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	120-180 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	120-180 min	100%						

2 Erster Studienabschnitt

7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.				
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Prof. Dr. Johannes Blömer			
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Modellierung:</i> Methodische Umsetzung Die Vorlesung nutzt Tafelanschrieb und Folien sowie kleine Aufgaben für die Studierenden während der Vorlesung. Sie wird sowohl durch Tafelübung als auch durch Kleingruppentutorien begleitet. Studierende haben in den Kleingruppen Gelegenheit, Aufgaben in der Gruppe zu bearbeiten und Übungsblätter durch Tutoren benoten zu lassen. Lernmaterialien, Literaturangaben <ul style="list-style-type: none"> • Uwe Kastens, Hans Kleine Büning, Modellierung • Angelika Steger, Diskrete Strukturen • Foliensatz der Vorlesung; Übungsblätter 			

Datenstrukturen und Algorithmen			
Data Structures and Algorithms			
Modulnummer: M.079.1211	Workload (h): 270	Leistungspunkte: 9	Turnus: Sommersemester
	Studiensemester: 2. Semester	Dauer (in Sem.): 1	Sprache: de

2 Erster Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.079.05201 Datenstrukturen und Algorithmen	V4 Ü2 Z1	105	135	P	400/25	
	b) L.079.05205 Praktikum: Datenstrukturen und Algorithmen	P	0	30	P	3	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Datenstrukturen und Algorithmen:</i> Empfohlene Vorkenntnisse: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse u. a. mit mathematischen Methoden zu erlernen <i>Teilnahmevoraussetzungen der Lehrveranstaltung Praktikum: Datenstrukturen und Algorithmen:</i> Empfohlene Vorkenntnisse: <ul style="list-style-type: none"> • Grundlagen der Programmierung 						

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Datenstrukturen und Algorithmen:</i> Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus "für den Rechner verstehbar". Algorithmen spielen daher eine zentrale Rolle in der Informatik. Wesentliches Ziel des Algorithmenentwurfs ist die (Ressourcen-) Effizienz, d.h. die Entwicklung von Algorithmen, die ein gegebenes Problem möglichst schnell oder mit möglichst geringem Speicherbedarf lösen. Untrennbar verbunden mit effizienten Algorithmen sind effiziente Datenstrukturen, also Methoden, große Datenmengen im Rechner so zu organisieren, dass Anfragen wie Suchen, Einfügen, Löschen aber auch komplexere Anfragen effizient beantwortet werden können. Die in dieser Veranstaltung vorgestellten Entwurfs- und Analysemethoden für effiziente Algorithmen und Datenstrukturen sowie die grundlegenden Beispiele wie Sortierverfahren, dynamische Suchstrukturen und Graphenalgorithmen gehören zu den wissenschaftlichen Grundlagen für Algorithmenentwicklung und Programmierung in weiten Bereichen der Informatik.</p> <ul style="list-style-type: none">• Einführung: Rechenmodelle, Effizienzmaße, Beispiele• Sortierverfahren: Quicksort, Heapsort, Mergesort• Datenstrukturen: Verkettete Listen, Bäume, Graphen, Dynamische Suchbäume, Hashing, Prioritätswarteschlangen• Entwurfs- und Analyseverfahren: Rekursion und das Mastertheorem, Teile-und-Herrsche, Dynamische Programmierung, Backtracking, Branch & Bound, Greedy Algorithmen• Graphenalgorithmen: Kürzeste Wege, Minimale Spannbäume <p><i>Inhalte der Lehrveranstaltung Praktikum: Datenstrukturen und Algorithmen:</i> Begleitend zur Vorlesung Datenstrukturen und Algorithmen werden in diesem Programmierpraktikum einige wichtige Algorithmen und Datenstrukturen exemplarisch implementiert. Studierende werden in konkreten Projekten das Problem analysieren, geeignete Programmiertechniken auswählen, praktisch realisieren und eine quantitative Leistungsbewertung durchführen.</p> <ul style="list-style-type: none">• Sortieralgorithmen• Einfache Graph-Algorithmen• Algorithmen für Wege-Probleme• Algorithmen zur Berechnung minimaler Spannbäume• Wörterbücher• Hashing
---	--

2 Erster Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden kennen effiziente Datenstrukturen und Algorithmen für ausgewählte grundlegende Probleme. Sie sind in der Lage:</p> <ul style="list-style-type: none"> • Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen einzusetzen • selbstständig und kreative Algorithmen und Datenstrukturen entwickeln (wie gestalte ich den kreativen Prozess vom algorithmischen Problem zum effizienten Algorithmus?) • mathematischer Methoden zum Korrektheitsbeweis und zur Effizienzanalyse einzusetzen. • die Wechselwirkung zwischen Algorithmus und Datenstruktur an wesentlichen Beispielen erläutern. • die Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten einschätzen <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Einsatz und Engagement • Gruppenarbeit • Haltung und Einstellung • Lernkompetenz • Motivationale und volitionale Fähigkeiten • Selbststeuerungskompetenz 												
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td>a) - b)</td> <td>Klausur</td> <td>120-180 min</td> <td>100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a) - b)	Klausur	120-180 min	100%				
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote										
a) - b)	Klausur	120-180 min	100%										
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td>SL</td> </tr> <tr> <td>b)</td> <td>Praktikumsarbeit mit anschließendem Gespräch</td> <td></td> <td>QT</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung und die qualifizierte Teilnahme konkret zu erbringen sind.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL	b)	Praktikumsarbeit mit anschließendem Gespräch		QT
zu	Form	Dauer bzw. Umfang	SL / QT										
a)	Schriftliche Übungsaufgaben		SL										
b)	Praktikumsarbeit mit anschließendem Gespräch		QT										
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung und erfolgreiche Absolvierung des Praktikums Datenstrukturen und Algorithmen</p>												
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>												

2 Erster Studienabschnitt

10	<p>Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</p>
11	<p>Verwendung des Moduls in anderen Studiengängen: keine</p>
12	<p>Modulbeauftragte/r: Prof. Dr. Sevag Gharibian</p>
13	<p>Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Datenstrukturen und Algorithmen:</i> Methodische Umsetzung:</p> <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafelanschrieb • Übungen in Kleingruppen • erwartete Aktivitäten der Studierenden: aktive Mitarbeit bei Präsenzübungen, Hausaufgaben • Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt • In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt. <p>Lernmaterialien, Literaturangaben:</p> <ul style="list-style-type: none"> • Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter <p><i>Hinweise der Lehrveranstaltung Praktikum: Datenstrukturen und Algorithmen:</i> Methodische Umsetzung:</p> <ul style="list-style-type: none"> • Programmieraufgaben in kleineren Teams <p>Lernmaterialien, Literaturangaben:</p> <ul style="list-style-type: none"> • Aufgabenstellungen, man pages, eigenständige Recherche zu weiterführender Literatur

Berechenbarkeit und Komplexität			
Computability and Complexity			
Modulnummer: M.079.1212	Workload (h): 180	Leistungspunkte: 6	Turnus: Wintersemester
	Studiensemester: 3. Semester	Dauer (in Sem.): 1	Sprache: de

2 Erster Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.079.05301 Berechenbarkeit und Komplexität	V3 Ü2	75	105	P	200/30	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Berechenbarkeit und Komplexität:</i> Erfolgreicher Abschluss der Module Modellierung und Datenstrukturen und Algorithmen						
4	Inhalte: <i>Inhalte der Lehrveranstaltung Berechenbarkeit und Komplexität:</i> Einführung in grundlegende Methoden und Techniken zur Charakterisierung der Schwierigkeit von Berechnungsproblemen. Als formales Rechenmodell werden Turingmaschinen definiert. Ausgehend hiervon werden die wichtigsten Begriffe und Techniken der Berechenbarkeitstheorie (wie z. B. Entscheidbarkeit, Unentscheidbarkeit, Diagonalisierung, Reduktionen) und der Komplexitätstheorie (wie z. B. Zeitkomplexität, Klassen P und NP, NP-Vollständigkeit, polynomielle Reduktionen) definiert und erläutert. <ul style="list-style-type: none"> • Einführung: Sprachen, Rechenmodelle, Grammatiken, Simulationen • Berechenbarkeit: Entscheidbare, unentscheidbare Sprachen, Diagonalisierung, Halteproblem, Reduktionen, Beispiele • Zeitkomplexität: Laufzeiten, Klassen P und NP, polynomielle Reduktionen, NP-Vollständigkeit, SAT, Satz von Cook-Levin, Beispiele • Approximationsalgorithmen 						
5	Lernergebnisse und Kompetenzen: Studierende kennen wesentliche Konzepte und Methoden der Berechenbarkeitstheorie und der Komplexitätstheorie. Sie können selbständig Probleme analysieren und klassifizieren. Studierende können Hypothesen zur Komplexität von Problemen entwickeln und diese anschließend verifizieren oder falsifizieren und darauf aufbauend neue Hypothesen formulieren. Nichtkognitive Kompetenzen: <ul style="list-style-type: none"> • Haltung und Einstellung • Selbststeuerungskompetenz 						

2 Erster Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur	90-120 min	100%
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.			
7	Studienleistung, qualifizierte Teilnahme:		
zu	Form	Dauer bzw. Umfang	SL / QT
a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Prof. Dr. Johannes Blömer		
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Berechenbarkeit und Komplexität:</i> Methodische Umsetzung Die Vorlesung nutzt Tafelanschrieb und Folien sowie kleine Aufgaben für die Studierenden während der Vorlesung. Sie wird sowohl durch Tafelübung als auch durch Kleingruppentutorien begleitet. Studierende haben in den Kleingruppen Gelegenheit, Aufgaben in der Gruppe zu bearbeiten und Übungsblätter durch Tutoren benoten zu lassen. Lernmaterialien, Literaturangaben <ul style="list-style-type: none"> • Michael Sipser: Introduction to the Theory of Computation • Uwe Schöning: Theoretische Informatik - kurz gefasst • Foliensatz der Vorlesung; Übungsblätter 		

2.3 Computersysteme

2 Erster Studienabschnitt

Digitaltechnik						
Digital Design						
Modulnummer: M.079.1310	Workload (h): 150	Leistungspunkte: 5	Turnus: Sommersemester			
	Studiensemester: 2. Semester	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05200 Digitaltechnik	V2 Ü2	60	90	P	300/25
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Digitaltechnik:</i> Empfohlene Vorkenntnisse Kenntnisse aus der Lehrveranstaltung Modellierung sind hilfreich.					
4	Inhalte: <i>Inhalte der Lehrveranstaltung Digitaltechnik:</i> Die Veranstaltung gibt eine Einführung in den Entwurf digitaler Schaltungen und Systeme. Dabei wird der Bogen vom Logikentwurf auf Gatterebene bis hin zu komplexeren Systemen auf Register-Transfer-Ebene gespannt. Die vermittelten Techniken und Methoden werden in den Übungen an Beispielen vertieft mit modernen Entwurfswerkzeugen umgesetzt. Im einzelnen werden die folgenden Themen behandelt: <ul style="list-style-type: none"> • Darstellung von Information und Fehlerkorrigierende Codes • Boolesche Algebra • Gatter und Schaltnetze • Logikoptimierung (Optimierung zweistufiger Logik nach Quine/McCluskey) • Automaten und Schaltwerke (festverdrahtet, mikroprogrammierbar) • Arithmetische Einheiten als Entwurfsbeispiele • Entwurf auf Register-Transfer-Ebene • Hardware-Beschreibungssprachen und Entwurf mit VHDL 					

2 Erster Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sind in der Lage</p> <ul style="list-style-type: none"> • den Entwurfsablauf in der Digitaltechnik von der Spezifikation bis zur technischen Realisierung zu beschreiben, • die zugrunde liegenden mathematischen Modelle aus der Booleschen Algebra und der Automatentheorie anzuwenden, • digitale Schaltungsentwürfe im Hinblick auf vorgegebene Entwurfsziele zu analysieren, • einfache digitale Systeme selbständig zu konzipieren, sowie • einfache digitale Systeme mit den entsprechenden Entwurfswerkzeugen technisch zu realisieren. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">60-90 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	60-90 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	60-90 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <p>keine</p>								
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>keine</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr. Marco Platzner</p>								

2 Erster Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Digitaltechnik:</i></p> <p>Methodische Umsetzung</p> <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafel • Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer • Praktische Übungen zum Hardware-Entwurf (Teamarbeit) <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Vorlesungsfolien und Übungsblätter • Aufgabenblätter und technische Dokumentation für die Rechnerübungen • J. F. Wakerly, „Digital Design,“ 4th Edition, Upper Saddle River, NJ: Pearson / Prentice Hall, 2007 • Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien im jeweiligen PANDA-Kurs
----	--

Rechnerarchitektur						
Computer Architecture						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.1311	150	5	Wintersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	3. Semester	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05302 Rechnerarchitektur	V2 Ü2	60	90	P	300/25
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	<i>Teilnahmevoraussetzungen der Lehrveranstaltung Rechnerarchitektur:</i>					
	Empfohlene Vorkenntnisse					
	Kenntnisse aus der Lehrveranstaltung Digitaltechnik sind hilfreich.					

2 Erster Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Rechnerarchitektur:</i> Die Vorlesung gibt eine Einführung in den Aufbau und Entwurf moderner Rechensysteme. Insbesondere wird vermittelt, wie durch ein effizientes Zusammenspiel von Hardware und Software kostengünstige und leistungsstarke Rechner entwickelt werden können. Die vorgestellten Techniken und Methoden werden in den Übungen an Beispielen vertieft.</p> <ul style="list-style-type: none"> • Grundstrukturen, von Neumann Rechner • Befehlssätze und Assemblerprogrammierung • Leistungsbewertung • Datenpfad und Steuerung • Pipelining • Speicherhierarchie, insbesondere Cache-Management und virtueller Speicher • Ein-/Ausgabe 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,</p> <ul style="list-style-type: none"> • den Aufbau eines modernen Rechners sowie das Zusammenspiel von Hardware und Software zu beschreiben, • die zugrunde liegenden allgemeinen Entwurfsprinzipien und -strategien zu erklären und anzuwenden, • Rechnersysteme im Hinblick auf Leistung und Kosten zu analysieren und zu bewerten, sowie • selbständig einfache Assemblerprogramme zu schreiben. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">60-90 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	60-90 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	60-90 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <p>keine</p>								
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>keine</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								

2 Erster Studienabschnitt

10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Prof. Dr. Marco Platzner
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Rechnerarchitektur:</i> Methodische Umsetzung <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafelanschrieb • Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer • Rechnerübungen zur Assemblerprogrammierung Lernmaterialien, Literaturangaben <ul style="list-style-type: none"> • Vorlesungsfolien und Übungsblätter • D. A. Patterson, J. L. Hennessy: Computer Organization & Design – The Hardware / Software Interface (3rd Edition); Morgan Kaufmann, 2007; ISBN: 978-0-12-370606-5, ISBN-10: 0-12-370606-8 • Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien auf der Webseite und in den Vorlesungsfolien

Systemsoftware und systemnahe Programmierung						
System software and system-level programming						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.1312	270	9	Sommersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	4. Semester	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05401 Systemsoftware und systemnahe Programmierung	V4 Ü2 Z1	105	135	P	200/25
b)	L.079.05402 Praktikum: Systemsoftware und systemnahe Programmierung	P	0	30	P	3

2 Erster Studienabschnitt

2	<p>Wahlmöglichkeiten innerhalb des Moduls:</p> <p>keine</p>
3	<p>Teilnahmevoraussetzungen:</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Systemsoftware und systemnahe Programmierung:</i></p> <p>Empfohlene Vorkenntnisse Es ist dringend zu empfehlen, die Vorlesungen Programmierung und Modellierung erfolgreich abgeschlossen zu haben. Ebenso sollten Grundlagen der Rechnerarchitektur bekannt sein.</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Praktikum: Systemsoftware und systemnahe Programmierung:</i></p> <p>Empfohlene Vorkenntnisse Grundlagen der Programmierung</p>
4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Systemsoftware und systemnahe Programmierung:</i> Einführung in grundlegende Probleme, Aufgaben, Herausforderungen und Herangehensweisen für systemnahe Software (z. B. Betriebssysteme, Protokollstacks). Es wird ein konzeptioneller Zugang gewählt (anstelle eines beispielorientierten Ansatzes); besonderer Wert wird auf praktisch orientierte Programmierübungen in kleinen Projekten gelegt, die den selbständigen Umgang mit der Materie vertiefen.</p> <ul style="list-style-type: none"> • Prozessmodell, Prozesswechsel • Scheduling • Zusammenhang mit grundlegenden Aspekten der Rechnerarchitektur (z. B. Interrupt) • Prozesskoordination (z. B. Semaphore) • Betriebsmittelverwaltung (z. B. Deadlocks, Bankier-Algorithmus) • Speicherverwaltung (virtueller Speicher, Paging, ...) • Interprozesskommunikation, Client/Server • Grundlagen Rechnernetze, Internet • Systemnahe Programmierung (z. B. Socket-Schnittstelle, Thread-Bibliotheken) <p><i>Inhalte der Lehrveranstaltung Praktikum: Systemsoftware und systemnahe Programmierung:</i> Begleitend zur Vorlesung Systemsoftware und systemnahe Programmierung werden in diesem Programmierpraktikum Techniken der systemnahen Programmierung praktisch erprobt und eingeübt. Studierende werden in konkreten Projekte das Problem analysieren, geeignete Programmier-techniken auswählen, praktisch realisieren und eine quantitative Leistungsbewertung durchführen.</p> <p>Mögliche Projektthemen sind:</p> <ul style="list-style-type: none"> • Thread-Programmierung: Nutzung bestehender Bibliotheken; Realisierung einer User-Level Thread-Bibliothek • Socket-Programmierung: Implementierung einfacher Socket-Kommunikation; insbesondere Leistungsbewertung unterschiedlicher Varianten • Scheduling-Algorithmen implementieren • Automatisierung durch Shell-Skripte

2 Erster Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Studierende können Aufgabenstellungen der Systemsoftware identifizieren, unterschiedliche Ansätze zu Problemlösungen benennen, klassifizieren und unterscheiden, deren Vor- und Nachteile evaluieren und für ein Problem eine geeignete Lösung auswählen. Sie sind in der Lage, diese Verfahren in eigenen Anwendungen gezielt zum Einsatz zu bringen (bspw. Semaphoren zur Koordination nebenläufiger Aktivitäten).</p> <p>Studierende können ggf. neue Lösungen konstruieren (bspw. Scheduling-Strategie) und deren Leistungsfähigkeit systematisch durch Einsatz geeigneter (mathematischer oder informatischer) Werkzeuge analysieren, deren Eignung evaluieren und mit Alternativen kontrastieren.</p> <p>Praktische Erfahrung im Umgang mit und Einsatz von systemnaher Werkzeuge und Programmier-techniken.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz • Selbststeuerungskompetenz 												
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td>a) - b)</td> <td>Klausur</td> <td>120-180 min</td> <td>100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a) - b)	Klausur	120-180 min	100%				
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote										
a) - b)	Klausur	120-180 min	100%										
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td>a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td>SL</td> </tr> <tr> <td>b)</td> <td>Praktikumsarbeit</td> <td></td> <td>QT</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung und die qualifizierte Teilnahme konkret zu erbringen sind.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL	b)	Praktikumsarbeit		QT
zu	Form	Dauer bzw. Umfang	SL / QT										
a)	Schriftliche Übungsaufgaben		SL										
b)	Praktikumsarbeit		QT										
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>												
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>												
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</p>												
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>												

2 Erster Studienabschnitt

12	Modulbeauftragte/r: Dr. Florian Klingler
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Systemsoftware und systemnahe Programmierung:</i> Methodische Umsetzung Die Vorlesung ist überwiegend folienorientiert, mit begleitendem Tafelinsatz und Aufgaben für die Studierenden während der Vorlesung. Sie wird sowohl durch Tafelübung als auch durch Kleingruppentutorien begleitet. Studierende haben in den Kleingruppen Gelegenheit, Aufgaben in der Gruppe zu bearbeiten und Übungsblätter durch Tutoren benoten zu lassen. Lernmaterialien, Literaturangaben Standardlehrbücher (z. B. Stallings, Betriebssysteme); Foliensatz der Vorlesung; Übungsblätter. <i>Hinweise der Lehrveranstaltung Praktikum: Systemsoftware und systemnahe Programmierung:</i> Methodische Umsetzung <ul style="list-style-type: none"> • Programmieraufgaben in kleineren Teams. Lernmaterialien, Literaturangaben <ul style="list-style-type: none"> • Aufgabenstellung, man pages, eigenständige Recherche zu unterstützender Kommunikation.

IT-Sicherheit			
IT Security			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.1313	150	5	Wintersemester
	Studiensemester:	Dauer (in Sem.):	Sprache:
	5. Semester	1	de
1	Modulstruktur		
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)
	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05500 IT-Sicherheit	V2 Ü2	60
		90	P
			150/25
2	Wahlmöglichkeiten innerhalb des Moduls:		
	keine		
3	Teilnahmevoraussetzungen:		
	<i>Teilnahmevoraussetzungen der Lehrveranstaltung IT-Sicherheit:</i>		
	Empfohlene Vorkenntnisse:		
	Rechnernetze, Programmierung, Systemsoftware und systemnahe Programmierung		

2 Erster Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung IT-Sicherheit:</i> In der Vorlesung werden die wesentlichen Begriffe und Probleme der IT Sicherheit vorgestellt. Es werden klassische und moderne Angriffstechniken auf Netzwerkprotokolle, Passwort-Datenbanken, Computersysteme und Webanwendungen vorgestellt und geeignete Gegenmaßnahmen diskutiert. Hierzu gehört auch die Vorstellung praxisrelevanter kryptographischer Protokolle und Algorithmen sowie deren Sicherheitseigenschaften. In der Veranstaltung werden folgende Themen behandelt:</p> <ul style="list-style-type: none"> • Ausgewählte Sicherheitsprobleme • Grundlagen der angewandten Kryptographie • Sicherheitsziele und Sicherheitsmodelle • Netzwerksicherheit • Anwendungssicherheit 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Studierende verstehen die wesentlichen Konzepte, Methoden und Mechanismen zum Schutz von Daten und Systemen vor Manipulation und Missbrauch auf einem grundlegenden, praxisorientierten, wissenschaftlichen Niveau. Sie sind in der Lage, die Konzepte zur Erhöhung der Systemsicherheit korrekt einzusetzen, einfache Sicherheitsprotokolle zu entwickeln und diese zu bewerten. Sie verstehen die Ursachen von Sicherheits-Problemen heutiger Systeme, sind in der Lage, grundlegende Konzepte auch in neuen Anwendungskontexten einzusetzen und besitzen ein generelles Bewusstsein für mögliche Sicherheitsbedrohungen und Risiken.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Einsatz und Engagement • Lernkompetenz • Lernmotivation 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">60-90 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	60-90 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	60-90 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						

2 Erster Studienabschnitt

8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Prof. Dr.-Ing. Juraj Somorovsky
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung IT-Sicherheit:</i> Methodische Umsetzung: <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafelanschrieb • Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer • Praktische Übungen zur IT Sicherheit Lernmaterialien, Literaturangaben: <ul style="list-style-type: none"> • Vorlesungsfolien und Übungsblätter • Sicherheit und Kryptographie im Internet, Jörg Schwenk • Computer Security, William Stallings und Lawrie Brown

2.4 Mathematik

Analysis für Informatiker			
Calculus for Computer Science			
Modulnummer: M.105.9710	Workload (h): 240	Leistungspunkte: 8	Turnus: Wintersemester
	Studiensemester: 1. Semester	Dauer (in Sem.): 1	Sprache: de

2 Erster Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.105.96100 Analysis für Informatiker	V4 Ü2	90	150	P	450/40	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: keine						
4	Inhalte: <i>Inhalte der Lehrveranstaltung Analysis für Informatiker:</i> Die in diesen Veranstaltungen vermittelten Kenntnisse werden als Verfahren bzw. Faktenwissen im Informatikstudium gebraucht; bzw. die mathematisch-methodische Denkweise (Definition, Satz, Beweis), die hier eingeübt wird. Kapitel I Grundbegriffe <ol style="list-style-type: none"> 1. Mengen und Abbildungen 2. Vollständige Induktion und Rekursion, Kombinatorik 3. Elementare Zahlentheorie 4. Reelle Zahlen, Körper 5. Die komplexen Zahlen Kapitel II Analysis <ol style="list-style-type: none"> 1. Konvergenz von Folgen 2. Konvergenz von Reihen und Potenzreihen 3. Stetigkeit 4. Exponentialfunktion und trigonometrische Funktionen 5. Polarkoordinaten, Einheitswurzeln und der Fundamentalsatz der Algebra 6. Differenzierbarkeit 7. Lokale Extrema, Taylor-Formel, Taylor-Reihen 8. Integrierbarkeit (Riemann-Integral) 9. Approximation von Nullstellen und Fixpunkten. Das Newton-Verfahren 						

2 Erster Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> • beschreiben den progressiven Aufbau des Zahlensystems (bis hin zu den komplexen Zahlen) und argumentieren mit dem Permanenzprinzip als formaler Leitidee, • verwenden die Begriffe der Konvergenz von Folgen und Reihen sowie der Vollständigkeit der reellen Zahlen formal sicher und erläutern diese Begriffe an tragenden Beispielen, • beschreiben die Begriffe Stetigkeit und Differenzierbarkeit anschaulich und formal und begründen zentrale Aussagen über stetige und differenzierbare Funktionen, verwenden die Idee der Approximation durch Potenzreihen zur Beschreibung von Funktionen, • definieren den Begriff des Integrals formal und verwenden ihn in mathematischen Zusammenhängen, interpretieren das Integrieren als Flächenmessung und als Mittelwertbildung, • erläutern und begründen den Hauptsatz der Differential- und Integralrechnung, • nutzen Software zur Darstellung und Exploration mathematischer Modellierungen und als heuristisches Werkzeug zur Lösung von Anwendungsproblemen, • kennen und reflektieren Fragen der Umsetzung numerischer Verfahren auf dem Computer (z.B. Komplexität, Genauigkeit). <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernkompetenz • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">120-180 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	120-180 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	120-180 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Form</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</p>								

2 Erster Studienabschnitt

11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Prof. Dr. Jürgen Klüners
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Analysis für Informatiker:</i> Methodische Umsetzung Die Studierenden <ul style="list-style-type: none"> • präsentieren und erklären mathematische Sachverhalte • denken konzeptionell, analytisch und logisch • denken und handeln eigenständig • erarbeiten sich interessengelenkt selbstständig mathematische Einsichten

Lineare Algebra für Informatiker						
Linear Algebra for Computer Science						
Modulnummer: M.105.9720	Workload (h): 240	Leistungspunkte: 8	Turnus: Sommersemester			
	Studiensemester: 2. Semester	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.105.96200 Lineare Algebra für Informatiker	V4 Ü2	90	150	P	300/30
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: keine					

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Lineare Algebra für Informatiker:</i> Einführung in die Grundlagen der Linearen Algebra, die während des Informatikstudiums benötigt werden. Die Lineare Algebra thematisiert auf unterschiedlichen begrifflichen Ebenen praktisch und theoretisch das Lösen linearer Gleichungssysteme und darüber hinausgehend das Konzept der Linearität als universell einsetzbares mathematisches Lösungswerkzeug. Dessen Rolle für das weitere Studium liegt in der großen Bedeutung, welche die Linearisierung (oder Lineare Approximation) für alle Sparten der Mathematik, für die mathematische Modellbildung und für die mathematischen Anwendungen hat.</p> <ol style="list-style-type: none">1. Grundbegriffe2. Vektorräume3. lineare Abbildungen4. Basis5. Dimension6. Matrizen7. lineare Gleichungssysteme8. Determinanten9. Eigenwerte10. charakteristisches Polynom11. Normalformenproblem
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden</p> <ul style="list-style-type: none">• verstehen und erläutern, wie abstrakte Vektorräume als koordinatenfreie Verallgemeinerung ein- bis dreidimensionaler Räume zustande kommen, und geben Beispiele aus der Mathematik und Anwendungsgebieten an, die in diesem konzeptionellen Rahmen verstanden werden können,• begreifen lineare Abbildungen von Vektorräumen als strukturverträgliche Abbildungen und erläutern, wie lineare Gleichungssysteme koordinatenfrei durch sie beschrieben werden,• verstehen den abstrakten Basis- und Dimensionsbegriff und erklären, wie dieser als Verallgemeinerung des naiven Koordinaten- und Dimensionsbegriff verstanden werden kann,• stellen lineare Abbildungen durch Matrizen dar und begreifen diese als koordinatenabhängige Realisierung,• verstehen und erläutern, wie sich die (eindeutige) Lösbarkeit solcher Gleichungssysteme charakterisieren lässt; lösen lineare Gleichungssysteme und erklären Lösungsverfahren,• verstehen die Determinante als alternierende Multilinearform und erläutern sie anhand ihrer geometrischen Bedeutung; begreifen ihre Rolle für die Inversion von Matrizen und kennen die Verfahren zu ihrer Bestimmung,• kennen den Begriff des Eigenwerts; verstehen und erklären das Normalformenproblem, kennen Kriterien für Diagonalisierbarkeit <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Lernkompetenz• Selbststeuerungskompetenz

2 Erster Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)			
	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
	a)	Klausur	120-180 min	100%
	Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.			
7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben		SL
	Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Prof. Dr. Jürgen Klüners			
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Lineare Algebra für Informatiker:</i> Methodische Umsetzung Die Studierenden <ul style="list-style-type: none"> • reflektieren eigene Lernerfahrungen • präsentieren und erklären mathematische Sachverhalte • denken konzeptionell, analytisch und logisch • erarbeiten sich interessengeleitet selbständig neue Erkenntnisse • denken und handeln eigenständig 			

Stochastik für Informatiker
Stochastic for Computer Science

2 Erster Studienabschnitt

Modulnummer: M.105.9730	Workload (h): 180	Leistungspunkte: 6	Turnus: Wintersemester			
	Studiensemester: 3. Semester	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.105.96300 Stochastik für Informatiker	V3 Ü2	75	105	P	200/25
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: Erfolgreicher Abschluss mindestens eines der Module Analysis für Informatiker oder Lineare Algebra für Informatiker					

4	<p>Inhalte:</p> <p>Deskriptive Statistik und Datenanalyse, Klassische Wahrscheinlichkeitsmodelle, Axiomatik, Standardverteilungen (u.a. Binomial), Satz von Bayes und Anwendungen, Bsp. für nicht-diskrete Ws.räume, Zufallsgrößen und ihre Momente, Quantile, Unabhängigkeit und mehrdimensionale Wahrscheinlichkeitsverteilungen, Gesetze der großen Zahlen, Zentraler Grenzwertsatz, Schätzen (inkl.-Konfidenzintervalle) und Testen, Simulation und Zufallszahlen, Markovketten, Unabhängigkeit und mehrdimensionale Wahrscheinlichkeitsverteilungen</p> <p><i>Inhalte der Lehrveranstaltung Stochastik für Informatiker:</i></p> <p>Deskriptive Statistik und Datenanalyse, Klassische Wahrscheinlichkeitsmodelle, Axiomatik, Standardverteilungen (u.a. Binomial), Satz von Bayes und Anwendungen, Bsp. für nicht-diskrete Wahrscheinlichkeitsräume, Zufallsgrößen und ihre Momente, Quantile, Gesetze der großen Zahlen, Zentraler Grenzwertsatz, Schätzen (inkl. Konfidenzintervalle) und Testen, Simulation und Zufallszahlen, Markovketten, mehrdimensionale Wahrscheinlichkeitsverteilungen</p> <p>Deskriptive Statistik und Datenanalyse</p> <ul style="list-style-type: none">• planen statistische Erhebungen (Befragung, Beobachtung oder Experiment), führen sie durch und werten sie aus• lesen und erstellen grafische Darstellungen für uni- und bivariate Daten (z.B. Kreuztabelle) und bewerten deren Eignung für die jeweilige Fragestellung• bestimmen und verwenden uni- und bivariate Kennwerte (z.B. Mittelwerte, Streumaße, Korrelationen, Indexwerte) und interpretieren sie angemessen <p>Zufallsmodellierung</p> <ul style="list-style-type: none">• modellieren mehrstufige Zufallsversuche durch endliche Ergebnismengen und nutzen geeignete Darstellungen (Baumdiagramm, Mehrfeldertafel)• rechnen und argumentieren mit Wahrscheinlichkeiten, bedingten Wahrscheinlichkeiten, Erwartungswerten und stochastischer Unabhängigkeit• erläutern inhaltlich das Bernoullische Gesetz der großen Zahlen und den zentralen Grenzwertsatz und deren Konsequenzen• verwenden diskrete und kontinuierliche Verteilungen und ihre Eigenschaften zur Modellierung <p>Stochastische Anwendungen</p> <ul style="list-style-type: none">• kennen Beispiele für die Anwendung von Stochastik in verschiedenen Wissenschaften (Ökonomie, Physik, ...)• schätzen in Zufallssituationen Parameter aus Daten• führen Hypothesentests durch und reflektieren deren zentralen Schritte und bestimmen Konfidenzintervalle• erläutern Unterschiede zwischen Bayes-Statistik und klassischen Testverfahren <p>Neue Medien</p> <ul style="list-style-type: none">• verwenden Tabellenkalkulation und statistische Software zur Darstellung und explorativen Analyse von Daten• simulieren Zufallsversuche computergestützt
---	---

2 Erster Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <ul style="list-style-type: none"> • Kenntnis der Bedeutung der Stochastik in Gesellschaft und Wissenschaft. • Sicherer Umgang mit den Begriffen der Stochastik in Wort und Schrift. • Verständnis des mathematischen Sachverhaltes und den damit verbundenen Denkweisen. • Verständnis der Beweise. Befähigung zur Lösung von Übungsaufgaben zur Stochastik. Fähigkeit des Erkennens von Verbindungen innerhalb der Stochastik beziehungsweise zwischen der Stochastik und anderen Bereichen der Mathematik. • Durchführung von einfachen statistischen Analysen. Befähigung zum Umgang mit einem Software-Paket zur Stochastik. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernkompetenz • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur</td> <td style="text-align: center;">90-120 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur	90-120 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur	90-120 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr. Martin Kolb</p>								

2 Erster Studienabschnitt

13	Sonstige Hinweise: keine
----	------------------------------------

3 Zweiter Studienabschnitt

Im zweiten Studienabschnitt haben die Studierenden die Gelegenheit, sich in Fächern ihrer Wahl gezielt zu vertiefen. Gleichzeitig wird weiterhin die Breite des Studiums garantiert, indem alle Studierenden mindestens eine weiterführende Veranstaltung aus den vier Gebieten Algorithmen und Komplexität, Computersysteme, Daten und Wissen sowie Softwaretechnik belegen müssen. Weiter gehören zu diesem Studienabschnitt ein Proseminar und die Bachelorarbeit.

3.1 Algorithmen und Komplexität

Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus für den Rechner verstehbar. Algorithmen spielen daher eine zentrale Rolle in der Informatik.

Deshalb steht im Mittelpunkt des Bachelormoduls Algorithmen und Komplexität die Klassifizierung von Problemen bezüglich ihrer algorithmischen Komplexität. Als Maße für Komplexität werden insbesondere Laufzeit und Speicherbedarf, aber auch z.B. Parallelisierbarkeit herangezogen. Module dieses Gebiets behandeln sowohl die Entwicklung und Analyse effizienter Algorithmen und algorithmischer Techniken, als auch die Untersuchung der Problem-inhärenten Komplexität, d.h. den Nachweis unterer Komplexitätsschranken und den Komplexitätsvergleich von Problemen. Weiter ergänzt wird das Gebiet durch ein Modul zur Kryptographie. Hier wird die inhärente Schwierigkeit von Problemen, wie sie die Komplexitätstheorie nachzuweisen versucht, positiv etwa für den Entwurf sicherer Verschlüsselungsverfahren genutzt.

Algorithmische Geometrie			
Computational Geometry			
Modulnummer: M.079.2215	Workload (h): 180	Leistungspunkte: 6	Turnus: Sommersemester
	Studiensemester: 6	Dauer (in Sem.): 1	Sprache: de

3 Zweiter Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.079.05601 Algorithmische Geometrie	V3 Ü2	75	105	WP	40/20	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	<p>Teilnahmevoraussetzungen:</p> <p>Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Algorithmische Geometrie:</i></p> <p>Empfohlene Vorkenntnisse Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen mathematische Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analyse wird angenommen.</p>						
4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Algorithmische Geometrie:</i> Es werden Algorithmen und Datenstrukturen aus dem Bereich der Algorithmischen Geometrie behandelt. Die Grundelemente der Eingabe sind geometrische Daten (Punkte, Linien, Kreise, Polygone, Körper). Die Probleme werden geometrisch formuliert und dafür wird eine algorithmische Lösung mit Hilfe spezieller geometrischer Datenstrukturen gesucht. Die Algorithmen werden theoretisch analysiert. Dazu wird Laufzeit und Speicherplatz bestimmt und die Korrektheit der Algorithmen bewiesen.</p>						
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden können grundlegende Techniken im Bereich der algorithmischen Geometrie anwenden. Sie können entscheiden, für welches geometrische Problem welcher Algorithmus am besten geeignet ist. Sie können die Algorithmen an eine neue Situation anpassen.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Haltung und Einstellung • Lernkompetenz • Schreib- und Lesekompetenz (wissenschaftlich) • Selbststeuerungskompetenz 						

3 Zweiter Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.			
7	Studienleistung, qualifizierte Teilnahme:		
zu	Form	Dauer bzw. Umfang	SL / QT
a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Dr. Matthias Fischer		

3 Zweiter Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Algorithmische Geometrie:</i></p> <p>Methodische Umsetzung</p> <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafelanschrieb • Übungen in Kleingruppen • erwartetet Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben • Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt • In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen vorgestellt. <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Standardlehrbücher, Folien der Vorlesung, Übungsblätter • Computational Geometry: Algorithms and Applications, Mark de Berg, Otfried Cheong, Marc van Krefeld, Marc Overmars, Springer-Verlag, 2008 • Algorithmische Geometrie, Rolf Klein, Springer-Verlag, 2005 • Lectures on Discrete Geometry, Jiri Matousek, Springer Verlag, 2001
----	--

Einführung in Kryptographie						
Introduction to Cryptography						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.2210	180	6	Wintersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	5	1	en			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05503 Einführung in Kryptographie	V3 Ü2	75	105	WP	100/25
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.					
	<i>Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Kryptographie:</i>					
	Empfohlene Vorkenntnisse					
	Datenstrukturen und Algorithmen sowie Berechenbarkeit und Komplexität					

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Einführung in Kryptographie:</i> In dieser Vorlesung werden die wichtigsten Aufgaben und Methoden der modernen Kryptographie vorgestellt. Weiter werden einige der wichtigsten Sicherheitsanforderungen moderner Kryptographie informell diskutiert. Es werden die Vor- und Nachteile symmetrischer und asymmetrischer Kryptographie erläutert. Wichtige kryptographische Basiskonstruktionen wie Verschlüsselungsverfahren und digitale Signaturen werden vorgestellt.</p> <ul style="list-style-type: none"> • Aufgaben der Kryptographie • Symmetrische und asymmetrische Verfahren • Elementare Sicherheitskonzepte und Kryptanalyse • Symmetrische Verschlüsselungsverfahren - DES, AES • Asymmetrische Verschlüsselungsverfahren - RSA, Elgamal • Schlüsselaustauschverfahren - Diffie-Hellman • Hashfunktionen und MACs - SHA3 • Digitale Signaturen - RSA, Elgamal, DSA 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Studierende sind in der Lage Sicherheitsanforderungen mittels kryptographischer Aufgaben zu formulieren. Sie kennen die wichtigsten kryptographischen Basistechniken und ihre Einsatzmöglichkeiten. Studierende können einschätzen, ob umgesetzte kryptographische Lösungen gegebenen Anforderungen genügen und sie können für gegebene Sicherheitsanforderungen die geeigneten kryptographischen Verfahren auszuwählen. Studierende können einschätzen, welche Anpassungen an kryptographische Verfahren unproblematisch sind und welche sicherheitskritisch sind.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernkompetenz • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						

3 Zweiter Studienabschnitt

8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).
11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Prof. Dr. Johannes Blömer
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Einführung in Kryptographie:</i> Methodische Umsetzung Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. Lernmaterialien, Literaturangaben Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography, Chapman and Hall, Johannes Buchmann: Einführung in Kryptographie, Springer Verlag, A Graduate Course in Applied Cryptography: https://crypto.stanford.edu/~dabo/cryptobook/ , Vorlesungsfolien, Übungsaufgaben

Grundlegende Algorithmen			
Fundamental Algorithms			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2211	180	6	Wintersemester
	Studiensemester:	Dauer (in Sem.):	Sprache:
	5	1	en
1	Modulstruktur		
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)
	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05511 Grundlegende Algorithmen	V3 Ü2	75
		105	WP
			100/30
2	Wahlmöglichkeiten innerhalb des Moduls:		
	keine		

3 Zweiter Studienabschnitt

3	<p>Teilnahmevoraussetzungen:</p> <p>Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Grundlegende Algorithmen:</i></p> <p>Empfohlene Vorkenntnisse</p> <p>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.</p>
4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Grundlegende Algorithmen:</i></p> <p>In dieser Veranstaltung werden die folgenden algorithmischen Paradigmen untersucht, einschließlich genauer Analysen von Laufzeit und Korrektheit:</p> <ul style="list-style-type: none">• Fortgeschrittene Suchstrukturen: Bereichssuche, Splay-Bäume, (a,b)-Bäume• Prioritätswarteschlangen und Anwendungen: Fortgeschrittene Heap-Implementierungen, Anwendungen• Graphenalgorithmen: Zusammenhangskomponenten, kürzeste Wege, Matchings• Netzwerkflüsse: Algorithmus von Ford und Fulkerson, Preflow-Push-Algorithmus, Anwendungen• Lineare Programmierung: Geometrische Interpretation, Dualität, Anwendungen• String Matching-Algorithmen, Knuth-Morris-Pratt-Algorithmus, Boyer-Moore-Algorithmus
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden wenden Entwurfsmethoden für effiziente Datenstrukturen und Algorithmen für schwierige Probleme wie Matching, Netzwerk-Fluß u.a. an. Sie nutzen mathematisch fundierte Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen. Darüber hinaus entwickeln sie selbstständig, kreativ Algorithmen und Datenstrukturen (Wie gestalte ich den kreativen Prozess vom algorithmischen Problem zum effizienten Algorithmus?) unter Nutzung von Entwurfsmethoden und ihrem Verständnis für die Struktur des algorithmischen Problems. Zudem nutzen sie einfache Varianten von fortgeschrittenen algorithmische Modellen wie online, approximative oder randomisierte Algorithmen.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Haltung und Einstellung• Selbststeuerungskompetenz

3 Zweiter Studienabschnitt

6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%;">zu</th> <th style="width: 50%;">Form</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr. Sevag Gharibian</p>								
13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Grundlegende Algorithmen:</i></p> <p>Methodische Umsetzung</p> <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafelanschrieb. • Übungen in Kleingruppen. • erwartete Aktivitäten der Studierenden: aktive Mitarbeit bei Präsenzübungen, Hausaufgaben. • Übungsblätter, Lösungen werden in Übungsgruppen vorgestellt und diskutiert. • In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt. <p>Lernmaterialien, Literaturangaben</p> <p>Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter</p>								

3 Zweiter Studienabschnitt

Komplexitätstheorie						
Complexity Theory						
Modulnummer: M.079.2212	Workload (h): 180	Leistungspunkte: 6	Turnus: Wintersemester			
	Studiensemester: 5	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05512 Komplexitätstheorie	V3 Ü2	75	105	WP	30
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.					
4	Inhalte: <i>Inhalte der Lehrveranstaltung Komplexitätstheorie:</i> Die Komplexitätstheorie ist eine wichtige Ergänzung der Theorie der Algorithmen. Ihr Ziel ist es zu verstehen, warum gewisse Berechnungsprobleme schwierig sind und diese anhand ihrer Schwierigkeit zu klassifizieren. Das bekannteste und wichtigste Beispiel ist die Theorie der NP-Vollständigkeit. <ul style="list-style-type: none"> • Komplexitätsklassen, P vs. NP • Reduktionen und Vollständigkeit • Platzkomplexität • Hierarchiesätze • Relativierung und Orakel-Turingmaschinen • Polynomialzeit-Hierarchie • Probabilistische Komplexitätsklassen 					

3 Zweiter Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Studierende können die Eigenschaften wesentlicher Komplexitätsklassen wie P, NP und PSPACE benennen und erläutern. Studierende können Probleme in geeignete Komplexitätsklassen einordnen. Studierende beherrschen die wichtigsten Techniken, um Probleme gemäß ihrer Komplexität vergleichen zu können (Reduktionen). Sie können diese Techniken auf neue Probleme anwenden.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Einsatz und Engagement • Lernmotivation • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Form</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr. Johannes Blömer</p>								

3 Zweiter Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Komplexitätstheorie:</i></p> <p>Methodische Umsetzung Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft.</p> <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Michael Sipser, Introduction to the Theory of Computation, • S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press, • Vorlesungsfolien, Übungsaufgaben
----	---

Verteilte Algorithmen und Datenstrukturen						
Distributed Algorithms and Data Structures						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.2214	180	6	Sommersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	6	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05610 Verteilte Algorithmen und Datenstrukturen	V3 Ü2	75	105	WP	60/30
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.					

4	<p>Inhalte:</p> <p>Die Vorlesung gibt eine Einführung in die Grundlagen der verteilten Algorithmen und Datenstrukturen. Einen Schwerpunkt bilden dabei hochskalierbare Datenstrukturen, die auch für sehr große und dynamische verteilte Systeme anwendbar sind. Nach einer Einführung in die Netzwerktheorie werden zunächst grundlegende Designprinzipien für verteilte Algorithmen und Datenstrukturen vorgestellt wie z.B. das Konzept der selbststabilisierenden Systeme. Danach folgt eine kurze Einführung in verteilte Programmierung, damit die in der Vorlesung vorgestellten Datenstrukturen auch implementiert werden können. Anschließend werden zunächst prozessorientierte Datenstrukturen und dann informationsorientierte Datenstrukturen vorgestellt.</p> <ul style="list-style-type: none">• Einleitung• Netzwerktheorie• Designprinzipien für verteilte Algorithmen und Datenstrukturen• Einführung in verteilte Programmierung• Prozessorientierte Datenstrukturen: (Zyklische) Listen, de Bruijn Graphen, Skip Graphen, Delaunay Graphen• Informationsorientierte Datenstrukturen: Verteiltes Hashing, verteilte Queue, verteilter Stack, verteilter Heap <p><i>Inhalte der Lehrveranstaltung Verteilte Algorithmen und Datenstrukturen:</i></p> <p>Die Vorlesung gibt eine Einführung in die Grundlagen der verteilten Algorithmen und Datenstrukturen. Einen Schwerpunkt bilden dabei hochskalierbare Datenstrukturen, die auch für sehr große und dynamische verteilte Systeme anwendbar sind. Nach einer Einführung in die Netzwerktheorie werden zunächst grundlegende Designprinzipien für verteilte Algorithmen und Datenstrukturen vorgestellt wie z.B. das Konzept der selbst-stabilisierenden Systeme. Danach folgt eine kurze Einführung in verteilte Programmierung, damit die in der Vorlesung vorgestellten Datenstrukturen auch implementiert werden können. Anschließend werden zunächst prozessorientierte Datenstrukturen und dann informationsorientierte Datenstrukturen vorgestellt.</p> <ul style="list-style-type: none">• Einleitung• Netzwerktheorie• Designprinzipien für verteilte Algorithmen und Datenstrukturen• Einführung in verteilte Programmierung• Prozessorientierte Datenstrukturen: (Zyklische) Listen, de Bruijn Graphen, Skip Graphen, Delaunay Graphen• Informationsorientierte Datenstrukturen: Verteiltes Hashing, verteilte Queue, verteilter Stack, verteilter Heap
---	--

3 Zweiter Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <ul style="list-style-type: none"> • Kenntnis ausgewählter verteilter Algorithmen und Datenstrukturen • Kenntnis wesentlicher Konzepte im Bereich verteilter Algorithmen und Datenstrukturen • Fähigkeit, selbstständig adäquate Techniken und Verfahren im Bereich der verteilten Algorithmen und Datenstrukturen zu entwickeln • Fähigkeit, algorithmische Probleme gemäß ihrer Lösbarkeit und Komplexität einzuschätzen • Fähigkeit, grundlegende verteilte Datenstrukturen zu implementieren <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz • Schreib- und Lesekompetenz (wissenschaftlich) • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Form</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr. Christian Scheideler</p>								

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Verteilte Algorithmen und Datenstrukturen:</i></p> <p>Methodische Umsetzung Vorlesung mit Übungen, Hausaufgaben und Softwareprojekt</p> <p>Lernmaterialien, Literaturangaben Skript</p>
----	---

3.2 Computersysteme

Jede Informatikanwendung benötigt ein technisches System, auf der sie ausgeführt wird. Beispiele für solche Systeme sind auf bestimmte Anwendungsklassen zugeschnittene Computersysteme wie eingebettete Systeme oder verteilte Systeme. Wesentliche Komponenten von Computersystemen sind neben der Rechner-Hardware auch die Betriebssysteme eines Rechners oder die technische Infrastruktur für die Vernetzung mehrerer Rechner.

Die Eignung und Qualität eines Gesamtsystems hängt wesentlich davon ab, dass die Eigenheiten, Vorteile und Einschränkungen dieser technischen Systeme sinnvoll ausgenutzt werden. Zudem entstehen aus der Evolution von Infrastrukturen neue Anwendungsklassen und tragen damit wesentlich zur Innovation in der Informatik und ihren Anwendungen bei. Wesentliche Beispiele sind hier zweifellos das Internet oder der Mobilfunk samt Smartphones, deren tief greifender Einfluss kaum überschätzt werden kann.

Das Verständnis solcher Systeme und die Fähigkeit, solche Systeme gezielt zu benutzen und weiterzuentwickeln, ist daher eine wesentliche Kernkompetenz eines Informatikers. In diesem Gebiet werden diese Fertigkeiten erworben und vertieft. Das Gebiet baut insbesondere auf den Modulen Digitaltechnik, Rechnerarchitektur sowie Systemsoftware und systemnahe Programmierung auf. Die dortigen Grundlagen werden hier methodisch vertieft, die zielgeleitete Konstruktion technischer Systeme wird untersucht und Verfahren zur Bewertung der Leistungsfähigkeit und Eignung solcher Systeme abgeleitet. Dabei fokussieren die einzelnen Module, die innerhalb dieses Gebiets gewählt werden können, auf einzelne Teilbereiche von Computersystemen: Vernetzung von Rechnern, verteilte Systeme, Betriebssysteme und eingebettete Systeme.

Einführung in Hochleistungsrechnen			
Introduction to High-Performance Computing			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2318	180	6	Sommersemester
	Studiensemester:	Dauer (in Sem.):	Sprache:
	6	1	de

3 Zweiter Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) Einführung in Hochleistungsrechnen	V2 Ü3	75	105	WP	30	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt. <i>Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Hochleistungsrechnen:</i> Die Vorlesung baut auf den Inhalten der Lehrveranstaltungen “Rechnerarchitektur” und “Systemsoftware und systemnahe Programmierung” auf.						
4	Inhalte: <i>Inhalte der Lehrveranstaltung Einführung in Hochleistungsrechnen:</i> Hochleistungsrechner sind eine zentrale Technologie für Modellierung, Simulation und Datenanalyse in zahlreichen wissenschaftlichen und technischen Anwendungen. Dieses Modul ist eine erste Einführung in das Hochleistungsrechnen und vermittelt einen Überblick über die Architektur, Leistungsbewertung, Programmierung und Optimierung paralleler Computersysteme. Im Fokus steht dabei die Erarbeitung eines übergreifenden Verständnisses von Hardwarearchitektur und Methoden der performance-orientierten Programmierung in Theorie und Praxis. Zur Vermittlung praktischer Fähigkeiten beinhaltet die Lehrveranstaltung auch einen erheblichen Anteil von Programmierübungen und Praktika auf Hochleistungsrechnern. Dabei werden sowohl Anwendungen aus dem Bereich der Computersimulation, wie auch aus dem Bereich der Datenanalyse betrachtet. Als Programmiersprache kommt die Sprache Julia zum Einsatz, die moderne dynamische Programmierparadigmen mit hoher Performanz vereint und besonders für numerische Berechnungen, Datenanalyse und Hochleistungsrechnen geeignet ist. Die Veranstaltung umfasst folgende Inhalte: <ul style="list-style-type: none"> • Übersicht über die Architektur von Parallel- und Hochleistungsrechnern • Leistungsbewertung von parallelen Programmen und Hochleistungsrechnern • Modelle und Programmiermuster für parallele Programmierung • Optimierung serieller Programme • Vektorisierung • Parallelisierung mit gemeinsamem Speicher (Multi-threading) • Parallelisierung mit verteiltem Speicher • Anwendungen und Fallstudien • Beschleuniger (GPU) 						

3 Zweiter Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,</p> <ul style="list-style-type: none"> • die grundlegenden Ansätze zur Modellierung, Leistungsbewertung und Optimierung von Rechnersystemen zu erklären und auf Fallstudien anzuwenden, • die Ebenen von Parallelverarbeitung in modernen Rechnersystemen zu benennen, ihre Eigenschaften zu charakterisieren und daraus Optimierungsstrategien abzuleiten, • Programmiertechniken zur Parallelverarbeitung beim Entwurf und der Implementierung von Software praktisch anzuwenden, • Parallele Anwendungen auf Hochleistungsrechnern auszuführen und eine Leistungsbewertung durchzuführen. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz • Selbststeuerungskompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <p>keine</p>								
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>keine</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr. Christian Plessl</p>								

3 Zweiter Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Einführung in Hochleistungsrechnen:</i></p> <p>Methodische Umsetzung In der Vorlesung werden die Grundlagen, Begriffe und Methoden des Hochleistungsrechnens vermittelt und mit Programmierbeispielen veranschaulicht. In den begleitenden Übungen werden die theoretischen Grundlagen vertieft und im begleitenden Praktikum in Programmierübungen anhand praktischer Fallstudien und kleinen Projekten in Kleingruppen selbst erprobt.</p> <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Vorlesungsfolien • Programmierbeispiele • Webseite
----	---

Eingebettete Systeme						
Embedded Systems						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.2311	180	6	Sommersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	6	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05620 Eingebettete Systeme	V3 Ü2	75	105	WP	50/25
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	<p>Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Eingebettete Systeme:</i></p> <p>Empfohlene Vorkenntnisse: Kenntnisse aus der Lehrveranstaltung Rechnerarchitektur sind hilfreich.</p>					

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Eingebettete Systeme:</i></p> <p>Die Veranstaltung bietet eine Einführung in Eingebettete Systeme und vermittelt Grundlagen zu Spezifikationsmodellen, eingebetteten Zielarchitekturen und Methoden zum Entwurf von reaktiven und Echtzeitsystemen sowie zur Bewertung und Analyse von Prozessor-Performance und -Energie.</p> <ul style="list-style-type: none"> • Einführung in Eingebettete Systeme • Spezifikationsmodelle: Zustandsorientiert, Datenflussorientiert • Zielarchitekturen: General-Purpose Prozessoren, Digitale Signalprozessoren, Mikrocontroller, ASIPs, FPGAs und ASICs, System-on-Chip • Reaktive und Echtzeitsysteme: Tasksdefinitionen, Programmieransätze, Echtzeitscheduling, gemeinsam genutzte Ressourcen • Performance und Energie: Worst-case execution time analysis, Energiemetriken, Techniken zur Energieminimierung 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,</p> <ul style="list-style-type: none"> • die Eigenschaften eingebetteter Systeme zu benennen, • Spezifikationsmodelle für eingebettete Systeme mit ihren Eigenschaften zu erklären, • die Entwurfsziele und Eigenschaften wesentlicher Typen von eingebetteten Zielarchitekturen zu erklären, • Ansätze zur Programmierung von Echtzeitsystemen aufzuzählen, • Methoden zur Analyse von Echtzeiteigenschaften anzuwenden, • Methoden zur Bestimmung der Worst-case execution time anzuwenden und • die Bedeutung von Performance- und Energie-Metriken einzuschätzen. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						

3 Zweiter Studienabschnitt

7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.				
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Prof. Dr. Marco Platzner			
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Eingebettete Systeme:</i> Methodische Umsetzung: <ul style="list-style-type: none"> • Vorlesung mit Beamer und Tafelanschrieb • Interaktive Übungen im Hörsaal • Rechnerübungen mit eingebetteten Zielarchitekturen (DSP, ARM, FPGA) Lernmaterialien, Literaturangaben: <ul style="list-style-type: none"> • Vorlesungsfolien und Übungsblätter • Aufgabenblätter und technische Dokumentation für die Rechnerübungen • Peter Marwedel: Embedded System Design, Springer, 2011. • Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien 			

Rechnernetze			
Computer Networks			
Modulnummer: M.079.2312	Workload (h): 180	Leistungspunkte: 6	Turnus: Wintersemester
	Studiensemester: 5	Dauer (in Sem.): 1	Sprache: de

3 Zweiter Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.079.05501 Rechnernetze	V3 Ü2	75	105	WP	60/30	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt. <i>Teilnahmevoraussetzungen der Lehrveranstaltung Rechnernetze:</i> Empfohlene Vorkenntnisse Vorlesung Systemsoftware und systemnahe Programmierung oder vergleichbar.						
4	Inhalte: <i>Inhalte der Lehrveranstaltung Rechnernetze:</i> Die Vorlesung Rechnernetze behandelt konzeptionelle und technologische Grundlagen von Rechnernetzen/Internet; thematisch werden dabei die Ebenen 1-4 des ISO/OSI-Modells abgedeckt. Zusätzlich werden Ansätze und Werkzeuge zur quantitativen Untersuchung von Kommunikationsprotokollen behandelt. Die Vorlesung wird durch eine Tafelübung begleitet. <ul style="list-style-type: none"> • Physikalische Schicht: Signalausbreitung, Modulation, Shannon-Grenzen • Sicherungsschicht: ARQ, FEC, Framing. Medienzugriffsverfahren (Aloha, CSMA, CSMA/CD). • Netzwerkschicht: Routing als Graphproblem und als Netzproblem; Standardverfahren (Dijkstra, Bellmann-Ford); Routing vs. Forwarding; Fallstudie IP (longest prefix matching, BGP, ...) • Transportschicht: Überlastabwehr, Flusskontrolle, Fairness, Fallstudie TCP. • Beschreibung von Diensten und Protokollen; quantitative Analyse von Kommunikationsprotokollen (z.B. Aloha, Markov-Kette für CSMA, Durchsatz bei TCP). 						

3 Zweiter Studienabschnitt

5	<p>Lernergebnisse und Kompetenzen:</p> <p>Absolventen der Lehrveranstaltung</p> <ul style="list-style-type: none"> • können die wesentlichen Aufgaben bei Konstruktion und Bau eines Rechnernetzes benennen und wesentliche Architekturansätze beschreiben; • können unterschiedliche Lösungen für ein Problem aufzählen, deren Vor- und Nachteile herausfinden und sich, gemäß der Anforderungen, für eine Lösung entscheiden; • Schwachstellen existierender Lösungen identifizieren und neue Kommunikationsprotokolle entwickeln und deren Leistungsfähigkeit bewerten. <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Einsatz und Engagement • Lernkompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr style="background-color: #f2f2f2;"> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Form</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">SL / QT</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Schriftliche Übungsaufgaben oder Mitarbeit in Kleinprojekt</td> <td></td> <td style="text-align: center;">SL</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</p>	zu	Form	Dauer bzw. Umfang	SL / QT	a)	Schriftliche Übungsaufgaben oder Mitarbeit in Kleinprojekt		SL
zu	Form	Dauer bzw. Umfang	SL / QT						
a)	Schriftliche Übungsaufgaben oder Mitarbeit in Kleinprojekt		SL						
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>Bestehen der Studienleistung</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Dr. Florian Klingler, Prof. Dr. Christian Plessl, Prof. Dr. Marco Platzner</p>								

13	<p>Sonstige Hinweise:</p> <p>Die Lehrveranstaltung findet derzeit nur alle zwei Jahr statt und wechselt im Wintersemester mit der Veranstaltung Verteilte Systeme ab.</p> <p><i>Hinweise der Lehrveranstaltung Rechnernetze:</i></p> <p>Methodische Umsetzung Folienbasierte Vorlesung mit Tafelanschrieb, durch Übung begleitet. Übungen dabei sowohl konzeptionell/analytisch als auch mit praktischen Aufgaben.</p> <p>Lernmaterialien, Literaturangaben Folien, Standardlehrbücher (insbes. Tanenbaum, Rechnernetze), Übungsblätter.</p> <ul style="list-style-type: none"> • Die Veranstaltung lässt sich sehr gut mit der Veranstaltung Verteilte Systeme ergänzen. • In einigen Semestern (wenn sowohl Rechnernetze als auch Verteilte Systeme angeboten werden) findet die Veranstaltung halbsemestrig statt; in der zweiten Semesterhälfte die Veranstaltung Verteilte Systeme.
----	---

3.3 Daten und Wissen

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.

Dieses Gebiet kombiniert die Themen Daten und Wissen im Sinne einer modern ausgerichteten KI und vermittelt methodische Grundlagen intelligenter Systeme. Die Inhalte des Gebiets erstrecken sich von Aspekten der Wissensrepräsentation über die automatisierte Wissensverarbeitung bis zum Erwerb von Wissen aus Daten. Der effektive Umgang mit Daten durch systematische Organisation und Verarbeitung großer Datenbestände wird in Vorlesungen zum Thema Datenbanken und Informationssysteme erlernt. Ein weiteres Thema dieses Gebiets, die Computergraphik und grafische Datenverarbeitung, widmet sich einer speziellen Form von Daten, nämlich digitalen Bildern, abstrakten Beschreibungen großer 3D Szenen, sowie der visuellen Kommunikation zwischen Mensch und Maschine. Die wichtige Echtzeit-Verarbeitung oft extrem umfangreicher grafischer Datenmengen wird durch die Implementierung von Code auf GPUs (Graphics Processing Units) vermittelt.

Databases and Information Systems
Databases and Information Systems

3 Zweiter Studienabschnitt

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.2412	180	6	Wintersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	5	1	en			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05532 Databases and Information Systems	V3 Ü2	75	105	WP	120/30
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	<p>Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.</p> <p><i>Teilnahmevoraussetzungen der Lehrveranstaltung Databases and Information Systems:</i></p> <p>Empfohlene Vorkenntnisse Studierende sollten Vorkenntnisse in relationalen Datenbanken und SQL haben, die in etwa denen der Vorlesung “Datenbanksysteme” entsprechen, sowie Vorkenntnisse im Programmieren, die in etwa denen der Vorlesungen “Programmierung” und “Grundlagen der Programmiersprachen” entsprechen.</p>					

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Databases and Information Systems:</i></p> <p>Datenspeicherung und Datenmanagement spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissens von Unternehmen in Daten abgelegt ist. Zudem wachsen die Mengen gespeicherter Daten ständig, und eine Verarbeitung dieser riesigen Datenmengen erfordert Kenntnisse, die über SQL und traditionelle Datenbanken hinausgehen. Beispiele für diese riesigen Datenmengen sind Genomdatenbanken, Textdokumentsammlungen, Sensordaten, Satellitendaten, Daten aus Kameras, Mikrofonen, oder RFID-Tags, Telekommunikationsdaten, Wetterdaten, Finanzdaten, Newsreader, Daten aus Messenger-Diensten, etc.. Anwendungen oder Informationssysteme zu entwickeln, die bei diesen Datenmengen akzeptable Antwortzeiten haben, erfordert Kenntnisse über Nicht-Standard-Datenmodelle, Hauptspeicher-Datenbanken, Kompression, Indizierung riesiger Datenbestände und effiziente Suche auf diesen Datenbeständen. Dieses Modul behandelt schwerpunktmäßig Algorithmen zur Kompression und zur effizienten Verarbeitung von komplexen, strukturierten Massendaten einschließlich Textdaten, Genomdaten, baumstrukturierter Daten und Graph-Daten. Zu den Inhalten gehört:</p> <ul style="list-style-type: none"> • Überblick über Suchmaschinen und Informationssysteme • Hauptspeicherdatenbanken und Succinct-Codierungstechniken • String-Kompressionsalgorithmen • Genom-Datenbanken • Verarbeitung riesiger Baum-Datenbestände (XML und JSON) und Baum-Kompression • Graph-Datenbanken und Graph-Kompression • Suchalgorithmen für Big Data und für Datenströme 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Nach Abschluss des Moduls sind die Studierenden in der Lage, XML-Verarbeitung in Softwaresystemen zu verstehen, zu entwerfen, zu implementieren und in Bezug auf ihren Zeit- und Platz-Bedarf zu beurteilen. Sie kennen wesentliche Such- und Anfragetechniken zur Informationsbeschaffung in unkomprimierten oder komprimierten XML-Datenbeständen. Sie sind in der Lage, unendliche Datenströme geeignet zu verarbeiten. Die Studenten sind in der Lage, sich neueste Forschungsergebnisse anhand von wissenschaftlichen Papers zu erarbeiten.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Gruppenarbeit • Lernkompetenz • Lernmotivation 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						

3 Zweiter Studienabschnitt

7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.				
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Prof. Dr. Stefan Böttcher			
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Databases and Information Systems:</i> Methodische Umsetzung Grundlegende Konzepte werden in einer Vorlesung präsentiert. Zusätzlich werden theoretische Konzepte in Tutorien in Kleingruppen vertieft, insbesondere für Kernkonzepte von Datenbanken wie die Suche in und Anfragen auf Big Data, verteilten Datenbanken und mobilem Datenmanagement. Zudem erwerben Studierende praktische Kenntnisse durch Computer-gestützte Übungen, in denen sie aufbauend auf den in der Vorlesung erläuterten Konzepten ihre eigenen Informationssysteme, Such- oder Kompressionsalgorithmen entwickeln. Lernmaterialien, Literaturangaben Verweise auf aktuelles Lernmaterial werden in der Vorlesung gegeben.			

Einführung in Human-Centered Machine Learning			
Introduction to Human-Centered Machine Learning			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2418	180	6	Sommersemester
	Studiensemester:	Dauer (in Sem.):	Sprache:
	6	1	en

3 Zweiter Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) Einführung in Human-Centered Machine Learning	V2 Ü3	75	105	WP	50	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Human-Centered Machine Learning:</i> Empfohlene Vorkenntnisse Grundkenntnisse in linearer Algebra, Statistik und Analysis werden vorausgesetzt.						
4	Inhalte: <i>Inhalte der Lehrveranstaltung Einführung in Human-Centered Machine Learning:</i> Dieser Kurs führt in Algorithmen des maschinellen Lernens ein, die in Kombination mit menschenzentrierten Daten verwendet werden können. Es werden sowohl grundlegende überwachte und unüberwachte Algorithmen des maschinellen Lernens als auch komplexere Deep-Learning-Ansätze, wie generative Modelle, vorgestellt. Alle Algorithmen werden im Zusammenhang mit menschenzentrierten Anwendungen vorgestellt, wie z. B. Alters- und Geschlechtsbestimmung oder der biometrischer Gesichtserkennung. Anschließend werden verschiedene Methoden zur Erzielung von Verantwortlichkeit in maschinellen Lernmodellen im Zusammenhang mit biometrischen Erkennungsmodellen vorgestellt. Diese zielen darauf ab, die Modelle datenschutzfreundlicher, fairer, zuverlässiger und erklärbarer zu machen. Die Veranstaltung umfasst folgende Inhalte: <ul style="list-style-type: none">• Überwachte Klassifizierung/Regression• Unüberwachtes Repräsentationslernen• Biometrie und Soft-Biometrie• Biometrische Systeme, Funktionsweisen und Evaluation• Gesichtserkennung• Fairness• Erklärbarkeit und Vertrauensschätzung• Präsentationsangriffe und Erkennung• Multimodale Fusion						
5	Lernergebnisse und Kompetenzen: Die Studierenden sind in der Lage <ul style="list-style-type: none">• verschiedene Modelle des maschinellen Lernens zu diskutieren und zu implementieren, die den Menschen in den Mittelpunkt stellen,• diese Modelle selbstständig zu evaluieren,• menschenzentrierte maschinelle Lernmodelle zu trainieren,• diese Modelle datenschutzfreundlicher, fairer, erklärbarer und zuverlässiger zu machen.						

3 Zweiter Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.			
7	Studienleistung, qualifizierte Teilnahme:		
zu	Form	Dauer bzw. Umfang	SL / QT
a)	Referat		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Dr.-Ing. Philipp Terhörst		

3 Zweiter Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Einführung in Human-Centered Machine Learning:</i></p> <p>Methodische Umsetzung</p> <p>Der Kurs führt in das Thema des maschinellen Lernens mit grundlegenden überwachten und unüberwachten maschinellen Lernansätzen ein. Diese Ansätze werden im Kontext von menschenzentrierten Anwendungen vorgestellt und diskutiert. Parallel zur Vorlesung werden die theoretischen Konzepte in den Übungen anhand von menschenzentrierten Daten eingeübt. Dies geschieht in Form von kurzen handschriftlichen und implementierten Aufgaben. Es wird eine Studienleistung in Form eines Kurzreferates erbracht.</p> <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning. Springer Publishing Company, Incorporated. • Anil K. Jain, Patrick Flynn, and Arun A. Ross. 2010. Handbook of Biometrics (1st. ed.). Springer Publishing Company, Incorporated. • Weitere Literatur wird in der Veranstaltung bekanntgegeben.
----	--

Foundations of the Semantic Web						
Foundations of the Semantic Web						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.2414	180	6	Sommersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	6	1	en			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05630 Foundations of the Semantic Web	V2 Ü3	75	105	WP	24
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					
3	Teilnahmevoraussetzungen:					
	Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.					
	<i>Teilnahmevoraussetzungen der Lehrveranstaltung Foundations of the Semantic Web:</i>					
	Empfohlene Vorkenntnisse					
	Logik, Java					

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Foundations of the Semantic Web:</i></p> <p>Die Lehrveranstaltung setzt sich mit der Repräsentation von maschinenlesbarem Wissen im Web auseinander. Nach der Vermittlung von Grundlagen zum Web werden die Syntax und formale Semantik der Sprachen RDF, RDFS, OWL, SPARQL untersucht. Außerdem werden Algorithmen zur Wissensextraktion und -integration vorgestellt. Praktische Anwendungsbeispiele für Technologien des Semantic Web runden den Inhalt der Veranstaltung ab.</p> <ul style="list-style-type: none"> • RDF Syntax und Semantik • RDFS Syntax und Semantik • OWL Syntax und Semantik • SPARQL Syntax und Semantik • Wissensextraktion • Verknüpfung von Ressourcen • Linked Data • Anwendungen 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden verstehen die Relevanz des Semantic Web und können relevante Techniken zur Darstellung von Wissen beschreiben und auswählen. Sie können vorhandene Techniken beschreiben, um einfache Wissensgraphen aus unstrukturierten und halbstrukturierten Datenquellen zu extrahieren. Sie verstehen die Konzepte der Wissensintegration, können Alternativen benennen und ein für ein bestimmtes Problem geeignetes Verfahren auswählen. Auf dieser Grundlage können die Studierenden wissensbasierte Anwendungen und deren Lebenszyklen entwerfen, realisieren und evaluieren.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernkompetenz • Medienkompetenz • Schreib- und Lesekompetenz (wissenschaftlich) • Gruppenarbeit 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" data-bbox="277 1442 1417 1621"> <thead> <tr> <th data-bbox="277 1442 363 1541">zu</th> <th data-bbox="363 1442 975 1541">Prüfungsform</th> <th data-bbox="975 1442 1198 1541">Dauer bzw. Umfang</th> <th data-bbox="1198 1442 1417 1541">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td data-bbox="277 1541 363 1621">a)</td> <td data-bbox="363 1541 975 1621">Klausur oder mündliche Prüfung</td> <td data-bbox="975 1541 1198 1621">90-120 min bzw. 40 min</td> <td data-bbox="1198 1541 1417 1621">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						

3 Zweiter Studienabschnitt

7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.				
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Prof. Dr. Axel-Cyrille Ngonga Ngomo			
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Foundations of the Semantic Web:</i> Methodische Umsetzung: In den wöchentlichen Vorlesungen (2 SWS) werden neue Inhalte behandelt. Neben formalen Betrachtungen werden auch Anwendungen und entsprechende Einschränkungen der im Laufe des Kurses vorgestellten Sprachen und Methoden behandelt. Die Übungen (1 SWS) sind sowohl theoretischer als auch praktischer Natur. Die Lernenden sollen zeigen, dass sie die Konzepte verstanden haben und sie auf praktische Probleme anwenden können. Das Miniprojekt (2 SWS) vermittelt den Studierenden eine ganzheitliche Sicht, wie ein komplexes Problem mit Hilfe von Technologien des Semantic Web gelöst werden kann. Lernmaterialien, Literaturangaben: Folien, Übungsaufgaben			

Natural Language Processing with Deep Learning			
Natural Language Processing with Deep Learning			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2419	180	6	Wintersemester
	Studiensemester:	Dauer (in Sem.):	Sprache:
		1	en

3 Zweiter Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) K.079.22360 Natural Language Processing with Deep Learning	V3 Ü2	75	105	WP	40/20	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Natural Language Processing with Deep Learning:</i> keine						
4	Inhalte: <i>Inhalte der Lehrveranstaltung Natural Language Processing with Deep Learning:</i> <ul style="list-style-type: none"> • Typische Aufgaben und Datensätze der natürlichen Sprachverarbeitung (NLP) und ihre Bewertung • Warum ist NLP schwierig? • Auffrischung der mathematischen Grundlagen, Calculus, gradientenbasierte Optimierung, Backpropagation für beliebige Funktionen • Log-lineare Modelle und Textklassifikation • Tiefe neuronale Netze • Sprachmodelle und Word embeddings • Lernen statischer Word embeddings • Rekurrente neuronale Netze • Encoder-Decoder, Text Generation, Attention und autoregressive Modelle • Transformers • Self-Attention und BERT • Reine Decoder-Modelle und GPT • LLMs: Prompting und In-context Lernen • Nicht-funktionale Eigenschaften von generativen NLP-Modellen: Privacy und Ethik 						
5	Lernergebnisse und Kompetenzen: Die Studierenden <ul style="list-style-type: none"> • verstehen das Rückgrat des modernen NLP, wie z.B. Embeddings und Transformer-basierte Modelle • können die Fähigkeiten und Schwächen verschiedener Deep-Learning-Modelle im NLP kritisch beurteilen • verstehen verschiedene Tasks im NLP, deren Bewertung und Modellierungsannahmen • können verschiedene Modelle und Ansätze in Python implementieren und sammeln praktische Erfahrungen 						

3 Zweiter Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.			
7	Studienleistung, qualifizierte Teilnahme: keine		
8	Voraussetzungen für die Teilnahme an Prüfungen: keine		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).		
11	Verwendung des Moduls in anderen Studiengängen: keine		
12	Modulbeauftragte/r: Prof. Dr. Ivan Habernal		
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Natural Language Processing with Deep Learning:</i> Methodische Umsetzung Wir werden Vorlesungen und Übungen haben. Die Vorlesungen werden ziemlich interaktiv sein, da ich versuche, die Studierenden in Fragen und kritisches Denken einzubeziehen, während wir fortfahren. Ich werde die Folien für jede Vorlesung im Voraus hochladen, damit die Studierenden sie für ihre Notizen verwenden können. Jede Vorlesung enthält Links zu relevanten Forschungsarbeiten für diejenigen, die sich eingehender mit dem Thema befassen möchten. Die Vorlesungen sind überwiegend theoretisch, d. h. wir behandeln die wichtigsten Konzepte, Ideen und mathematischen Beschreibungen, aber nicht, wie man es in einem bestimmten Framework programmiert. In den praktischen Kursen werden wir genau das Gegenteil tun. Die Studierenden werden mit aktuellen Mainstream-Frameworks für Deep Learning in NLP, wie Pytorch oder Huggingface, experimentieren. Wir werden ein breites Spektrum an Komplexitäten abdecken, von der Programmierung eines einfachen neuronalen Netzes von Grund auf bis hin zur Verwendung eines vortrainierten Sprachmodells. Lernmaterialien, Literaturangaben Das Modul orientiert sich nicht an einem bestimmten Lehrbuch, sondern wir werden uns auf frei verfügbare, qualitativ hochwertige Bücher stützen, wie z.B. das Buch von Yoav Goldberg und andere; diese werden in der ersten Vorlesung im Detail vorgestellt.		

3.4 Softwaretechnik

Das Gebiet vermittelt einen breiten Überblick über die wichtigsten Konzepte, Notationen und Methoden der Softwaretechnik und ihrer formalen und mathematischen Grundlagen. Die vermittelten Kenntnisse sollen die Studierenden in die Lage versetzen, Softwaresysteme unter vorgegebenen technischen und ökonomischen Randbedingungen zu entwickeln. Darüber hinaus sollen die Studierenden das wissenschaftliche Handwerkszeug beherrschen, um sich im späteren Berufsleben in zukünftige Techniken einzuarbeiten.

Angriffssicherer Softwareentwurf						
Secure Software Engineering						
Modulnummer: M.079.2115	Workload (h): 180	Leistungspunkte: 6	Turnus: Sommersemester			
	Studiensemester: 6	Dauer (in Sem.): 1	Sprache: en			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05640 Angriffssicherer Softwareentwurf	V3 Ü2	75	105	WP	100
2	Wahlmöglichkeiten innerhalb des Moduls: keine					
3	Teilnahmevoraussetzungen: Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.					

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Angriffssicherer Softwareentwurf:</i></p> <p>Was braucht es, um Softwaresysteme angriffssicher zu entwickeln? Dies ist die Schlüsselfrage, der wir in dieser Veranstaltung auf den Grund gehen. Um sie zu beantworten ist es erforderlich, ein Verständnis der folgenden Kernbereiche des angriffssicheren Softwareentwurfs zu entwickeln: Bedrohungsmodellierung, sicheres Design, sichere Programmierung, Sicherheitsvalidierung, sicheres Deployment und sichere Wartung. Diese Bereiche werden in dieser Veranstaltung auf beispielorientierte Weise abgedeckt. Diskutiert werden aktuelle Techniken, die auf diese Bereiche anwendbar sind, sowie die Lektionen, die aus konkreten Sicherheitsvorfällen gelernt werden können.</p> <ol style="list-style-type: none"> 1. Einführung in die Bedrohungsmodellierung und -analyse 2. Buffer Overflows: Prinzip, Exploits und Gegenmaßnahmen 3. Andere Code Injection-Schwachstellen: Prinzipien, Exploits und Gegenmaßnahmen 4. Crypto: gängige Algorithmen und Fallstricke 5. Zugriffskontrolle in Java und Android 6. Informationsfluss und Nutzungskontrolle 7. Automatische Erkennung von Schwachstellen: Codeanalyse, Fuzz Testing, modellbasiertes Testen 8. Systematische Sicherheitsanalyse 9. Softwareaktualisierung und -wartung 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Ziel der Veranstaltung ist es, dass die Teilnehmer ein fundiertes Verständnis der allerwichtigsten Aspekte des angriffssicheren Softwareentwurfs erhalten. Das schließt die Fähigkeit ein, Bedrohungen von Softwaresystemen zu identifizieren und zu modellieren, um die gängigsten Klassen von Schwachstellen zu vermeiden, sowie Techniken und Werkzeuge zu identifizieren und anzuwenden, um das Einführen von Sicherheitsschwachstellen zu verhindern oder zu identifizieren.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Lernkompetenz • Lernmotivation 								
6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%;">zu</th> <th style="width: 50%;">Prüfungsform</th> <th style="width: 20%;">Dauer bzw. Umfang</th> <th style="width: 20%;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <p>keine</p>								
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>keine</p>								

3 Zweiter Studienabschnitt

9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).
11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Prof. Dr. Eric Bodden
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Angriffssicherer Softwareentwurf:</i> Lernmaterialien, Literaturangaben: Gary McGraw: Software Security: Building Security In (2006, Addison-Wesley Professional). Über UPB als e-book verfügbar.

Modellbasierte Softwareentwicklung			
Model-Based Software Engineering			
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2114	180	6	Wintersemester
	Studiensemester:	Dauer (in Sem.):	Sprache:
	5	1	de
1	Modulstruktur		
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)
	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05543 Modellbasierte Softwareentwicklung	V3 Ü2	75
	105	WP	75/30
2	Wahlmöglichkeiten innerhalb des Moduls:		
	keine		
3	Teilnahmevoraussetzungen:		
	Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.		

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Modellbasierte Softwareentwicklung:</i></p> <p>In der modellbasierten Softwareentwicklung steht das Modell einer Software im Mittelpunkt. Es wird dabei nicht nur zu Dokumentationszwecken, sondern auch zur Entwicklung selbst verwendet (auch modellgetriebene Softwareentwicklung genannt). Übliche modellbasierte Techniken beinhalten unter anderem den Entwurf von Modellierungssprachen anhand von statischer und dynamischer Semantik sowie Metamodellierung sowie die Anwendung der Modelle in Form von Modelltransformationen, oder auch zum Model Checking oder für das Reverse Engineering von Softwarearchitekturen.</p> <p>Den Trend zur modellbasierten und modellgetriebenen Softwareentwicklung kann man sowohl in der Forschung, als auch in der Praxis beobachten und stellt daher eine wichtige Grundlage für die Ausbildung eines Softwareentwicklers dar.</p> <ul style="list-style-type: none"> • Software-Modelle, u.A. mit UML • Views und Viewpoints • Metamodellierung • Statische und Dynamische Semantik • Modelltransformationen • Softwarearchitekturen • Praxiseinsatz der modellbasierten und modellgetriebenen Softwareentwicklung • aktuelle Trends und Forschungsthemen in der modellbasierten und modellgetriebenen Softwareentwicklung 								
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Die Studierenden sollen grundlegende Verfahren zur Konstruktion großer Softwaresysteme kennen und ihre Anwendung beherrschen. Sie sollen die Vor- und Nachteile von Spezifikationstechniken erfahren, die Notwendigkeit von Design erkennen und Modelle zur Verbesserung der Softwarequalität einsetzen können. Unter anderem wird auf das Paradigma des „Model Driven Development“ eingegangen, das einen wesentlichen Produktivitäts- und Qualitätsgewinn bei der Softwareentwicklung verspricht.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none"> • Einsatz und Engagement • Lernkompetenz 								
6	<p>Prüfungsleistung:</p> <p><input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						

3 Zweiter Studienabschnitt

7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben		SL
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.				
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Prof. Dr. Gregor Engels			
13	Sonstige Hinweise: <i>Hinweise der Lehrveranstaltung Modellbasierte Softwareentwicklung:</i> Methodische Umsetzung Vorlesung mit Beamer und praktische Rechnerübungen. Lernmaterialien, Literaturangaben <ul style="list-style-type: none"> • Ghezzi: Fundamentals of Software Engineering (Addison Wesley) • Reiko Heckel, Gabriele Taentzer: Graph Transformation for Software Engineers - With Applications to Model-Based Development and Domain-Specific Language Engineering. Springer 2020 • Marco Brambilla, Jordi Cabot, Manuel Wimmer: Model-Driven Software Engineering in Practice, Second Edition. Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers 2017 			

Systematische Entwicklung von VR/AR Anwendungen			
Systematic Development of VR/AR Applications			
Modulnummer: M.079.2118	Workload (h): 180	Leistungspunkte: 6	Turnus: Sommersemester
	Studiensemester:	Dauer (in Sem.): 1	Sprache: de

3 Zweiter Studienabschnitt

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a) L.079.05604 Systematische Entwicklung von VR/AR Anwendungen	V3 Ü2	75	105	WP	50/25	
2	Wahlmöglichkeiten innerhalb des Moduls: keine						
3	Teilnahmevoraussetzungen: Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.						

3 Zweiter Studienabschnitt

4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Systematische Entwicklung von VR/AR Anwendungen:</i></p> <p>Obwohl die Begriffe Virtual Reality (VR) und Augmented Reality (AR) oftmals noch immer mit Unterhaltung und Videospiele assoziiert werden, haben sich die vielfältig nutzbaren Anwendungsbereiche der Technologien auch in anderen Branchen wie der Industrie, der Bildung oder auch der Medizin herausgestellt, wodurch deren Einsatz seit Langem Thema aktueller Forschung ist. Während Virtual Reality die Vorstellung einer virtuellen Welt realisiert, in der sich der Anwender durch die Manipulation seiner Sinne präsent fühlt und vollkommen in eine virtuelle Welt eintaucht, ohne jeglichen Bezug zur Außenwelt zu haben, ist Augmented Reality eine Erweiterung der Realität. Das Erwerben von und interagieren mit digitalen Daten in einem realen Raum steht hier im Vordergrund. Die Vorlesung führt grundlegende Technologien und Paradigmen der Virtual Reality (VR) und Augmented Reality (AR) ein. In den ersten Wochen lernen die Studierenden Methoden des Trackings, der 3D/2D Anzeige und Interaktion mit virtuellen Objekten sowie aktuelle Anzeigeegeräte (z.B. Microsoft HoloLens oder Meta Quest) und Software (z.B. Unity) kennen. Die erlernten Konzepte, Methoden und Technologien werden in praktischen Übungsaufgaben vertieft. Zusätzlich werden die Studierenden in Kleingruppen ein Mini-Forschungsprojekt im Themenkomplex VR/AR planen und umsetzen. Die Studierenden erarbeiten dabei auf Basis der aktuellen Forschungslage eine Forschungsfrage, die sie mit Hilfe des zu entwickelnden Prototypen beantworten möchten.</p> <p>Inhalt:</p> <ul style="list-style-type: none">• Einführung, Entwicklung, Begriffsdefinitionen• VR/AR-Hardware (Ein- und Ausgabegeräte)• Wahrnehmungsaspekte von VR/AR• Computergrafik von virtuellen Welten• Tracking• Anzeige, Interaktion und Navigation• Immersion• Modellierung und Annotation• Software-Architekturen• Echtzeit-Aspekte• Evaluation von VR/AR Anwendungen• VR/AR Systeme aus der Forschung
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Das Ziel dieser Vorlesung ist die Vermittlung von grundlegenden Methoden zur systematischen Erstellung und Darstellung virtueller Umgebungen. Die Studierenden sind in der Lage Virtual Reality (VR) und Augmented Reality (AR) - Systeme zu analysieren und zu evaluieren und kennen die verschiedenen Technologien und deren Probleme. Sie haben fortgeschrittene Programmierkenntnisse zur Erstellung von VR und AR Szenen u.a. mittels Unity3D. Es werden unter anderem Virtual-Reality- und Augmented-Reality-Anwendungen aus wissenschaftlich-technischen und industriellen Gebieten vorgestellt.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Einsatz und Engagement• Lernkompetenz• Medienkompetenz• Gruppenarbeit

3 Zweiter Studienabschnitt

6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)			
	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
	Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.			
7	Studienleistung, qualifizierte Teilnahme:			
	zu	Form	Dauer bzw. Umfang	SL / QT
	a)	Schriftliche Übungsaufgaben oder Mitarbeit in Kleinprojekt		SL
	Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen: Bestehen der Studienleistung			
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote: Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).			
11	Verwendung des Moduls in anderen Studiengängen: keine			
12	Modulbeauftragte/r: Dr. Enes Yigitbas			

3 Zweiter Studienabschnitt

13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Systematische Entwicklung von VR/AR Anwendungen:</i></p> <p>Methodische Umsetzung</p> <ul style="list-style-type: none"> • Vorlesung mit praktischen Übungsaufgaben • Mini-Forschungsprojekt in Kleingruppen zur Erstellung und Evaluation von VR/AR Anwendungen <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • Ralf Dörner, Wolfgang Broll, Paul Grimm, Bernhard Jung (Hrsg.): Virtual und Augmented Reality (VR/AR). Springer Vieweg, 2013. • William R. Sherman, Alan B. Craig: Understanding Virtual Reality – Interface, Application, and Design. Elsevier Science, 2003. • Dieter Schmalstieg, Tobias Höllerer: Augmented Reality – Principles and Practice. Addison-Wesley, 2016. • Joseph J. LaVioal, Ernst Kruijff, Ryan P. Mc Mahan, Doug A. Bowman, Ivan Poupyrev: 3D User Interfaces – Theory and Practice. Pearson Education, 2017. • Toni Parisi. Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile. O’Reilly, 2015. • Jonathan Linowes: Unity Virtual Reality Projects – Explore the world of virtual reality by building immersive and fun VR projects using Unity 3D. Packt Publishing, 2015.
----	--

Systems Engineering						
Systems Engineering						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
M.079.2119	180	6	Wintersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	5	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) L.079.05504 Systems Engineering	V2 Ü3	75	105	WP	20
2	Wahlmöglichkeiten innerhalb des Moduls:					
	keine					

3 Zweiter Studienabschnitt

3	<p>Teilnahmevoraussetzungen:</p> <p>Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.</p>
4	<p>Inhalte:</p> <p><i>Inhalte der Lehrveranstaltung Systems Engineering:</i></p> <p>Maschinelle Intelligenz, Industrie 4.0 – dahinter verbergen sich Intelligente Technische Systeme (ITS): Komplexe Produkte aus dem Zusammenspiel von Ingenieurwissenschaften und Informations- und Kommunikationstechnologien. Die Entwicklung von ITS stellt Maschinenbauer, Informatiker und Wirtschaftsingenieure vor neue Herausforderungen, wird aber in Zukunft immer bedeutender sein.</p> <p>Die Vorlesung vermittelt zunächst ein umfassendes Verständnis von ITS und erläutert den Wandel von der Mechatronik hin zu Systemen mit inhärenter (Teil-)Intelligenz. Im weiteren Verlauf beschäftigen sich die Studierenden mit geeigneten Entwurfs- und Spezifikationstechniken für die Entwicklung von ITS, primär dem Systems Engineering. Hierzu bietet die Vorlesung eine Einführung in die grundlegenden Methoden und Architekturen für ITS, die anhand aktueller Beispiele erläutert werden. Methoden für Analyse und Test von ITS auf Basis einer Systemspezifikation zeigen abschließend, dass eine ganzheitliche Vorgehensweise (das Systems Engineering) beim Entwurf der entscheidende Erfolgsfaktor für die technischen Systeme von morgen ist.</p> <ul style="list-style-type: none">• Einführung in die Grundlagen von ITS• Historie des Systems Engineerings• Systemtheorie und Systemdenken• Prozesse und Richtlinien des Systems Engineerings• Hilfsmittel für die Planung und den Entwurf von ITS
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Ziel der Vorlesung ist die Vermittlung von Grundwissen des Systems Engineering und Intelligenten technischen Systemen (ITS). Die Studierenden sind in der Lage, Methoden des Systems Engineering anzuwenden und auf Beispiele von ITS anzuwenden. Wesentliche ist dabei die Aneignung von Wissen zum Thema Systemtheorie und der Anwendung des Systemdenkens, auf technische und nicht technische Systeme. Hierfür werden Richtlinien und Werkzeuge für die Planung und Entwicklung von ITS vorgestellt und angewendet.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Gruppenarbeit• Einsatz und Engagement

3 Zweiter Studienabschnitt

6	<p>Prüfungsleistung:</p> <p> <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP) </p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 10%; text-align: center;">zu</th> <th style="width: 50%; text-align: center;">Prüfungsform</th> <th style="width: 20%; text-align: center;">Dauer bzw. Umfang</th> <th style="width: 20%; text-align: center;">Gewichtung für die Modulnote</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">a)</td> <td>Klausur oder mündliche Prüfung</td> <td style="text-align: center;">90-120 min bzw. 40 min</td> <td style="text-align: center;">100%</td> </tr> </tbody> </table> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote	a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%
zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote						
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%						
7	<p>Studienleistung, qualifizierte Teilnahme:</p> <p>keine</p>								
8	<p>Voraussetzungen für die Teilnahme an Prüfungen:</p> <p>keine</p>								
9	<p>Voraussetzungen für die Vergabe von Credits:</p> <p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>								
10	<p>Gewichtung für Gesamtnote:</p> <p>Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</p>								
11	<p>Verwendung des Moduls in anderen Studiengängen:</p> <p>keine</p>								
12	<p>Modulbeauftragte/r:</p> <p>Prof. Dr.-Ing. Roman Dumitrescu</p>								
13	<p>Sonstige Hinweise:</p> <p><i>Hinweise der Lehrveranstaltung Systems Engineering:</i></p> <p>Lernmaterialien, Literaturangaben</p> <ul style="list-style-type: none"> • acatech (Hrsg.): Cyber-Physical Systems – Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion (acatech POSITION). Springer-Verlag, 2011 • Gausemeier, J.; Dumitrescu, R.; Steffen, D.; Czaja, A.; Wiederkehr, O.; Tschirner, C.: Systems Engineering in der industriellen Praxis. Paderborn, 2013 								

4 Weiteres

Schlüsselqualifikation						
Modulnummer: M.079.2510	Workload (h): 150	Leistungspunkte: 5	Turnus: Sommer- / Wintersemester			
	Studiensemester: beliebiges Semester	Dauer (in Sem.): 1	Sprache: de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) Proseminar	PS2	15	105	P	15
	b) Mentoring	Treffen in Kleingruppen	15	15	P	20
2	Wahlmöglichkeiten innerhalb des Moduls: Für das Proseminar können alle Proseminare aus dem Angebot des Bachelorstudiengangs Informatik gewählt werden.					
3	Teilnahmevoraussetzungen: <i>Teilnahmevoraussetzungen der Lehrveranstaltung Proseminar:</i> Empfohlene Vorkenntnisse Je nach gewähltem Thema.					

4 Weiteres

4	<p>Inhalte:</p> <p>Das Modul besteht aus zwei Teilen: dem Proseminar und dem Mentoring.</p> <p><i>Inhalte der Lehrveranstaltung Proseminar:</i></p> <p>Im Proseminar soll beispielhaft die Einarbeitung in ein wissenschaftliches Thema erlernt und abstraktes Denken gestärkt werden. Die Inhalte sollen schriftlich und mündlich präsentiert werden. Dazu soll Basiswissen in Bezug auf Literaturrecherche, Rhetorik und aktuelle Präsentationstechniken sowie in Bezug auf Kritikfähigkeit und Feedbackmethoden erworben und angewendet werden.</p> <p>Die vermittelte Inhaltskompetenz betreffs der fachlichen Ausrichtung des Proseminars ist abhängig vom jeweiligen Thema der Veranstaltung. Unabhängig davon werden den Studierenden im Proseminar Fakten zur Erstellung und dem Ablauf von Präsentationen (Medienkompetenz), sowie dem Umgang mit Literatur, sowie zur Anfertigung schriftlicher Ausarbeitungen vermittelt.</p> <p><i>Inhalte der Lehrveranstaltung Mentoring:</i></p> <p>Im Mentoring werden Studierende einzelnen Lehrenden und deren Mitarbeiterinnen bzw. Mitarbeitern in Mentorengruppen (15 – 20 Studierende) zugeordnet. Es finden während des gesamten Bachelorstudiums je nach Bedarf etwa zweimal im Semester Treffen statt. Ziel ist es, durch Beratung – individuell oder in Kleingruppen – Probleme des Studiums und des Faches zu bearbeiten. Dabei sollen Engagement, Motivation und Selbstständigkeit als Aspekte von Selbstkompetenz gestärkt werden. Das Mentoring zielt auf Vermeidung unnötig langer Studiendauern und auf Reduktion der Abbrecherquote.</p>
5	<p>Lernergebnisse und Kompetenzen:</p> <p>Im Proseminar wird neben dem inhaltlichen Aspekt vor allem das Aufbereiten eines Themas und seine Präsentation eingeübt. Die Studierenden lernen in der praktischen Durchführung das Erarbeiten eines Themas, das Treffen von Auswahlen, das Halten von Vorträgen, den Umgang mit Fragen und Diskussionsbeiträgen, sowie das Anfertigen von größeren schriftlichen Texten. Die erarbeiteten Kompetenzen im Proseminar bereiten das Bewältigen ähnlicher Situationen später im Studium (Seminar, Projektgruppe, Abschlussarbeit) und im Beruf (Präsentationen, Berichte) vor.</p> <p>Nichtkognitive Kompetenzen:</p> <ul style="list-style-type: none">• Einsatz und Engagement• Haltung und Einstellung• Lernkompetenz• Lernmotivation• Medienkompetenz• Motivationale und volitionale Fähigkeiten• Schreib- und Lesekompetenz (wissenschaftlich)• Selbststeuerungskompetenz

4 Weiteres

6	Prüfungsleistung:	<input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
	zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
	a) - b)	Seminarvortrag und schriftliche Ausarbeitung	45-60 min	100%
<p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>				
7	Studienleistung, qualifizierte Teilnahme:	keine		
8	Voraussetzungen für die Teilnahme an Prüfungen:	keine		
9	Voraussetzungen für die Vergabe von Credits:	Die Vergabe von Credits erfolgt, wenn die Prüfungsleistung bestanden und das Mentoring absolviert worden ist.		
10	Gewichtung für Gesamtnote:	Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).		
11	Verwendung des Moduls in anderen Studiengängen:	keine		
12	Modulbeauftragte/r:	Studiengangsbeauftragter Informatik		
13	Sonstige Hinweise:	<p><i>Hinweise der Lehrveranstaltung Proseminar:</i> Methodische Umsetzung Referate mit schriftlicher Ausarbeitung und Vortrag. Lernmaterialien, Literaturangaben Je nach gewähltem Thema.</p> <p><i>Hinweise der Lehrveranstaltung Mentoring:</i> Es finden während des gesamten Bachelorstudiums je nach Bedarf etwa zweimal im Semester Treffen statt, in Kleingruppen oder individuell.</p>		

Studium Generale – Bachelor			
General Studies – Bachelor			
Modulnummer:	Workload (h): 210	Leistungspunkte: 7	Turnus: Sommer- / Wintersemester
	Studiensemester: Beliebig	Dauer (in Sem.): 1	Sprache: de

4 Weiteres

1	Modulstruktur						
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)	
	a)	Studium Generale – Bachelor	V3 Ü2	75	135	P	30
2	Wahlmöglichkeiten innerhalb des Moduls: Beliebige Veranstaltungen außerhalb der Informatik können gewählt werden.						
3	Teilnahmevoraussetzungen: keine						
4	Inhalte: Eine beliebige Kombination von Veranstaltungen außerhalb der Informatik und im Umfang von maximal 7 LP muss gewählt werden. Nebenfach und Studium Generale haben einen Umfang von insgesamt 25 LP. Die angegebene Verteilung der LP auf Lehrveranstaltungen ist nur exemplarisch. <i>Inhalte der Lehrveranstaltung Studium Generale – Bachelor:</i> Abhängig von den gewählten Veranstaltungen						
5	Lernergebnisse und Kompetenzen: 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. Nichtkognitive Kompetenzen: <ul style="list-style-type: none"> • Einsatz und Engagement • Kooperationskompetenz • Medienkompetenz • Schreib- und Lesekompetenz (wissenschaftlich) 						
6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)						
	zu	Prüfungsform		Dauer bzw. Umfang		Gewichtung für die Modulnote	
	a)	Prüfung im Studium Generale				100%	
	Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.						

4 Weiteres

7	Studienleistung, qualifizierte Teilnahme:		
	zu	Form	Dauer bzw. Umfang
	a)	Qualifizierte Teilnahme im Rahmen des Studium Generale	QT
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die qualifizierte Teilnahme konkret zu erbringen ist.			
8	Voraussetzungen für die Teilnahme an Prüfungen:		
keine			
9	Voraussetzungen für die Vergabe von Credits:		
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.			
10	Gewichtung für Gesamtnote:		
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).			
11	Verwendung des Moduls in anderen Studiengängen:		
keine			
12	Modulbeauftragte/r:		
Studiengangsbeauftragter Informatik			
13	Sonstige Hinweise:		
keine			

Bachelor-Abschlussarbeit						
Bachelor Thesis						
Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:			
A.079.2910	450	15	Sommer- / Wintersemester			
	Studiensemester:	Dauer (in Sem.):	Sprache:			
	6	1	de			
1	Modulstruktur					
	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
	a) Bachelor-Abschlussarbeit – Arbeitsplan		15	75	P	1
	b) Bachelor-Abschlussarbeit		15	345	P	1
2	Wahlmöglichkeiten innerhalb des Moduls:					
keine						

4 Weiteres

3	Teilnahmevoraussetzungen: Erfolgreicher Abschluss aller Module des ersten Studienabschnitts sowie des Moduls Schlüsselqualifikation.		
4	Inhalte: <i>Inhalte der Lehrveranstaltung Bachelor-Abschlussarbeit – Arbeitsplan:</i> 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. <i>Inhalte der Lehrveranstaltung Bachelor-Abschlussarbeit:</i> Die Bachelorarbeit umfasst die Bearbeitung eines Themas mit schriftlicher Ausarbeitung und einer mündlicher Präsentation der Ergebnisse. Der Studierende soll zeigen, dass er innerhalb einer vorgegebenen Frist ein Thema der Informatik auf der Grundlage wissenschaftlicher Methoden bearbeiten kann. Die Aufgabe einer Bachelorarbeit kann beispielsweise die Entwicklung von Software, Hardware, eine Beweisführung oder eine Literaturrecherche umfassen.		
5	Lernergebnisse und Kompetenzen: Im Rahmen ihrer Abschlussarbeit bearbeiten die Studierenden ein Problem nach wissenschaftlichen Methoden innerhalb einer bestimmten Frist. Die im Zuge des Studiums erworbenen fachlich-methodischen sowie fachübergreifenden Kompetenzen sollen dazu entsprechend eingesetzt werden. Dazu gehören insbesondere auch die Strukturierung und Planung der einzelnen Arbeitsschritte sowie die Präsentation der Ergebnisse nach Abschluss der Arbeit. Nichtkognitive Kompetenzen: <ul style="list-style-type: none"> • Einsatz und Engagement • Lernmotivation • Motivationale und volitionale Fähigkeiten • Schreib- und Lesekompetenz (wissenschaftlich) • Selbststeuerungskompetenz 		
6	Prüfungsleistung: <input checked="" type="checkbox"/> Modulabschlussprüfung (MAP) <input type="checkbox"/> Modulprüfung (MP) <input type="checkbox"/> Modulteilprüfungen (MTP)		
	zu	Prüfungsform	Dauer bzw. Umfang
	a) - b)	Abschlussarbeit	100%
7	Studienleistung, qualifizierte Teilnahme: keine		
8	Voraussetzungen für die Teilnahme an Prüfungen: keine		
9	Voraussetzungen für die Vergabe von Credits: Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.		
10	Gewichtung für Gesamtnote: Das Modul wird mit 48 Credits gewichtet.		

4 Weiteres

11	Verwendung des Moduls in anderen Studiengängen: keine
12	Modulbeauftragte/r: Studiengangsbeauftragter Informatik
13	Sonstige Hinweise: Die Bachelorarbeit 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. Die Aufgabenstellung soll so gestaltet werden, dass sie einem Arbeitsaufwand von neun Wochen Vollzeitarbeit entspricht. Die Arbeit wird studienbegleitend erstellt und muss fünf Monate nach der Ausgabe abgegeben werden. Sie soll einen Umfang von in der Regel nicht mehr als 60 DIN A4-Seiten haben. <i>Hinweise der Lehrveranstaltung Bachelor-Abschlussarbeit – Arbeitsplan:</i> Methodische Umsetzung Direkte Absprache mit Betreuer. Lernmaterialien, Literaturangaben Je nach gewähltem Thema. <i>Hinweise der Lehrveranstaltung Bachelor-Abschlussarbeit:</i> Methodische Umsetzung Selbständiges Arbeiten unterstützt durch individuelle Betreuung Lernmaterialien, Literaturangaben Je nach gewähltem Thema.

5 Module im Wintersemester

• M.105.9710 Analysis für Informatiker	38
• A.079.2910 Bachelor-Abschlussarbeit	95
• M.079.1212 Berechenbarkeit und Komplexität	26
• M.079.2412 Databases and Information Systems	69
• M.079.2210 Einführung in Kryptographie	51
• M.079.2211 Grundlegende Algorithmen	53
• M.079.1313 IT-Sicherheit	36
• M.079.2212 Komplexitätstheorie	55
• M.079.2114 Modellbasierte Softwareentwicklung	82
• M.079.1210 Modellierung	20
• M.079.2419 Natural Language Processing with Deep Learning	77
• M.079.1111 Programmiersprachen	6
• M.079.1110 Programmierung	4
• M.079.1311 Rechnerarchitektur	31
• M.079.2312 Rechnernetze	66
• M.079.2510 Schlüsselqualifikation	91
• M.079.1113 Softwaretechnikpraktikum	15
• M.105.9730 Stochastik für Informatiker	43
• Studium Generale – Bachelor	93
• M.079.2119 Systems Engineering	88

6 Module im Sommersemester

• M.079.2215 Algorithmische Geometrie	48
• M.079.2115 Angriffssicherer Softwareentwurf	80
• A.079.2910 Bachelor-Abschlussarbeit	95
• M.079.1114 Datenbanksysteme	9
• M.079.1211 Datenstrukturen und Algorithmen	22
• M.079.1310 Digitaltechnik	28
• M.079.2318 Einführung in Hochleistungsrechnen	61
• M.079.2418 Einführung in Human-Centered Machine Learning	72
• M.079.2311 Eingebettete Systeme	64
• M.079.2414 Foundations of the Semantic Web	75
• M.079.1115 Gestaltung von Nutzungsschnittstellen	18
• M.105.9720 Lineare Algebra für Informatiker	41
• M.079.2510 Schlüsselqualifikation	91
• M.079.1112 Software Engineering	11
• Studium Generale – Bachelor	93
• M.079.2118 Systematische Entwicklung von VR/AR Anwendungen	84
• M.079.1312 Systemsoftware und systemnahe Programmierung	33
• M.079.2214 Verteilte Algorithmen und Datenstrukturen	58

7 Modules in English

- M.079.2412 Databases and Information Systems 69
- M.079.2414 Foundations of the Semantic Web 75
- M.079.2211 Fundamental Algorithms 53
- M.079.2210 Introduction to Cryptography 51
- M.079.2418 Introduction to Human-Centered Machine Learning 72
- M.079.2419 Natural Language Processing with Deep Learning 77
- M.079.2115 Secure Software Engineering 80

Erzeugt am 11. September 2023 um 18:15.