

# Modulhandbuch

zum

Vollzeit Studiengang

Bachelor Informatik

12.09.2019

# Inhaltsverzeichnis

<b>Mathematik 1 - MA1</b> . . . . .	1
<b>Einführung in die Programmierung - EPR</b> . . . . .	2
<b>Technische Grundlagen der Informatik 1 - TEI1</b> . . . . .	4
<b>Grundlagen der Betriebswirtschaft - BWL</b> . . . . .	5
<b>Technisches Englisch - ENG</b> . . . . .	6
<b>Mathematik 2 - MA2</b> . . . . .	7
<b>Technische Grundlagen der Informatik 2 - TEI2</b> . . . . .	9
<b>Algorithmen und Datenstrukturen - ALD</b> . . . . .	10
<b>Betriebssysteme - BSY</b> . . . . .	11
<b>Objektorientierte Anwendungsentwicklung - OOA</b> . . . . .	12
<b>Grundlagen des Marketing - MAR</b> . . . . .	13
<b>Statistik - STA</b> . . . . .	14
<b>Graphische DV und Bildverarbeitung - GRA</b> . . . . .	15
<b>Web-Engineering - WEB</b> . . . . .	16
<b>Theoretische Informatik - THI</b> . . . . .	17
<b>Datennetze und Datenübertragung - DNÜ</b> . . . . .	18
<b>Verteilte Systeme - VSY</b> . . . . .	19
<b>Datenbanksysteme - DBS</b> . . . . .	20
<b>Datennetzmanagement - DNM</b> . . . . .	21
<b>Interaktive Systeme - IAS</b> . . . . .	22
<b>Numerik - WPV</b> . . . . .	23
<b>Seminarmodul - WPS</b> . . . . .	24
<b>Projektmanagement - PRM</b> . . . . .	25
<b>Recht in der Berufspraxis des Ingenieurs - JUR</b> . . . . .	26
<b>Vorlesungsmodul 2 - WPV2</b> . . . . .	27
<b>Softwareengineering - SWE</b> . . . . .	29
<b>IT-Sicherheit - ITS</b> . . . . .	30
<b>Echtzeitsysteme - EZS</b> . . . . .	31
<b>Projektmodul IT-Mobile IT mit Android - ITS</b> . . . . .	32
<b>Rechtliche und gesellschaftliche Aspekte der Informatik - RGA</b> . . . . .	33
<b>Netzwerksicherheit - WPV2</b> . . . . .	34
<b>Logikprogrammierung und funktionale Programmierung - WPV2</b> . . . . .	35

**Praxisphase - PRX** ..... 36  
**Bachelorarbeit - BA** ..... 37  
**Kolloquium - KOL** ..... 38

<b>Modul</b>	<b>MA1 Mathematik 1</b>			<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	1. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	4	60	30	
<b>Übung</b>	2	30	60	
<b>Praktikum</b>				
<b>Arbeitsaufwand in Stunden</b>		90	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Mathematische Kenntnisse und Rechenfähigkeit auf dem Niveau der Fachhochschulreife				
<b>Prüfungsvorleistung:</b>				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Die Studierenden können Probleme mathematisch und in der formalen Sprache der Logik ausdrücken. Die Studierenden können die grundlegenden Methoden und Verfahren der diskreten Mathematik, und linearen Algebra selbstständig anwenden				
<b>Inhalte:</b> Diskrete Mathematik und allgemeine Grundlagen: Aussagen, Mengen und Relationen, mathematische Beweisverfahren, elementare Zahlentheorie, Rekursion, algebraische Strukturen, Lineare Algebra: Vektorräume, Matrizen, Gleichungssysteme, geometrische Anwendungen				
<b>Lehrmethoden:</b> Vorlesung mit Literatur zum Selbststudium; Rechnen von Aufgaben in den Übungen und als Hausübungen				
<b>Bezug zu anderen Fächern/Modulen:</b> Das Modul vermittelt in vielen Modulen des Studiengangs benötigte mathematische Kenntnisse zur Analyse und Lösung von Problemstellungen der Informatik.				
<b>Literatur:</b>				
<ul style="list-style-type: none"> <li>• Hartmann, P: Mathematik für Informatiker, Vieweg</li> <li>• Teschl, G. &amp; S.: Mathematik für Informatiker Bd. I (Diskrete Mathematik und lineare Algebra), Springer</li> <li>• Schubert, M.: Mathematik für Informatiker, Vieweg</li> <li>• Stingl: Mathematik für Fachhochschulen Technik und Informatik, Hanser</li> </ul>				
<b>Dozenten:</b> Tipp				
<b>Modulverantwortliche:</b> Tipp				
<b>Aktualisiert:</b> 15.05.2018				

<b>Modul</b>	<b>EPR Einführung in die Programmierung</b>		<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	1. Semester		inkl. Prüfungsvorbereitung
<b>Sem. Lehrveranstaltung</b>	4	60	60
<b>Praktikum</b>	2	30	30
<b>Arbeitsaufwand in Stunden</b>		90	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> keine			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> schriftliche benotete Prüfung (2 Stunden)			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls <ul style="list-style-type: none"> <li>• können kleinere Programme für vorgegebene Aufgabenstellungen in der Programmiersprache C erstellen,</li> <li>• kennen die Grundprinzipien von Algorithmen und können einfache Algorithmen entwickeln und in strukturierten Programmen umsetzen,</li> <li>• beherrschen das imperative Programmierparadigma,</li> <li>• kennen die mögliche Unterstützung durch Entwicklungsumgebungen,</li> <li>• besitzen die Fähigkeit zur Abstraktion,</li> <li>• können sich in vorhandene Programme einarbeiten und vorhandene Programmelemente sinnvoll nutzen.</li> </ul>			
<b>Inhalte:</b> Algorithmen. Einführung in die Programmiersprache C und Grundlagen der Strukturierten Programmierung: Ablaufstrukturen, Datentypen und Funktionen, einfache Datenstrukturen wie verkettete Listen. Außerdem: Elementare Ein- und Ausgabe, Dateisystem, Speicherverwaltung, rekursive Funktionen und Anwenden des Erlernten auf einfache Algorithmen.			
<b>Lehrmethoden:</b> Vorlesung mit integrierten Übungen am Rechner; Diskussion von Fachinhalten und Problemen; Praktikum in kleinen Gruppen an Rechnern; begleitendes eigenverantwortliches Lernen. Begleitend wird ein freiwilliges Programmierprojekt (em Erstsemesterprojekt) angeboten.			
<b>Bezug zu anderen Fächern/Modulen:</b> Das Modul vermittelt grundlegende Programmierkenntnisse, die in fast allen anderen Modulen benötigt werden.			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• J. Wolf: C von A bis Z. Galileo Computing.</li> <li>• B.W. Kernighan, D.M. Ritchie: Programmieren in C. Carl Hanser Verlag.</li> <li>• K.H. Zeiner: Programmieren lernen mit C. Hanser.</li> <li>• H. P. Gumm, M. Sommer: Einführung in die Informatik. Oldenboerg Verlag</li> <li>• Fibelkorn: Die schwarze Kunst der Programmierung. Semele Verlag.</li> </ul>			
<b>Dozenten:</b> Davids, Nitsche			
<b>Modulverantwortliche:</b> Davids			
<b>Aktualisiert:</b> 10.05.2018			

<b>Modul</b>	<b>TEI1 Technische Grundlagen der Informatik 1</b>			<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	1. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	3	45	30	
<b>Übung</b>	2	30	30	
<b>Praktikum</b>	1	15	30	
<b>Arbeitsaufwand in Stunden</b>		90	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Mathematik, physikalische Elektrotechnik (Schulkenntnisse)				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an diesem Modul sind die Studierenden in der Lage, die Grundlagen für die technische Realisierung der Booleschen Algebra nachzuvollziehen. Sie können aufgrund logischen und algorithmischen Denkens kombinatorische Schaltungen bis hin zur Arithmetic Logic Unit analysieren, entwerfen und realisieren. Als Grundlage für den Rechnerentwurf lernen Sie von einfachen digitalen Speicherelementen über Zähler bis hin zu komplexeren Schaltwerken (Modellierung und Implementierung als endliche Automaten) die notwendigen Komponenten kennen. Die Erlangung der Kompetenz zum Umgang mit Hardwarebeschreibungssprachen für den Entwurf und die Synthese digitaler Schaltungen ist ein weiteres wichtiges Lernziel</p>				
<p><b>Inhalte:</b> Grundlagen zur Digitaltechnik:</p> <ul style="list-style-type: none"> <li>• Zahlendarstellung und Codes</li> <li>• Boolesche Algebra</li> <li>• Schaltnetze</li> <li>• Vereinfachen von booleschen Gleichungen</li> <li>•</li> <li>• Digitale Schaltungen:</li> <li>• Technische Realisierung von Schaltungen</li> <li>• Zeitliches Verhalten von Schaltungen</li> <li>• Schaltungsbeschreibungssprachen</li> <li>• Standardschaltnetze</li> <li>• Schaltwerke und synchrone Schaltungen</li> <li>• Standardschaltwerke und endliche Automaten</li> <li>• Entwurf von synchronen Schaltungen mit Schaltungsbeschreibungssprachen</li> <li>•</li> <li>• Grundlagen zur Rechnerarchitektur:</li> <li>• Darstellung von Daten und Informationen</li> <li>• Information und Informationsgehalt</li> <li>• Optimaler Code</li> <li>• Fehlererkennung und -korrektur</li> <li>• Kodierung von reellen Zahlen</li> <li>• Fließkommaarithmetik</li> </ul>				
<p><b>Lehrmethoden:</b></p> <ul style="list-style-type: none"> <li>• Vorlesung mit Skript und Literatur zum Selbststudium</li> <li>• Rechnen von Aufgaben in Hausübungen und Vortrag in den Übungsstunden sowie Nachbereitung im Selbststudium</li> <li>• Praktische Lösung von Hausaufgaben anhand bereitgestellter FPGA-Boards</li> <li>• Vorbereitung der Laborarbeit im Selbststudium</li> <li>• Aufbau digitaler Schaltungen im Labor</li> <li>• selbständige Durchführung einer (nicht vorbereiteten) Tagesaufgabe im Labor</li> </ul>				
<b>Bezug zu anderen Fächern/Modulen:</b> Voraussetzung für das Modul "Technische Grundlagen der Informatik II"				

**Literatur:**

- Skript zur Vorlesung
- Tocci: Digital Systems. New Jersey: Pearson Prentice Hall, 2004
- Floyd: Digital Fundamentals. New Jersey: Pearson Prentice Hall, 2005
- Hoffmann: Grundlagen der Technischen Informatik. Carl Hanser Verlag München, 2007
- Himpe: Digitale Logik selbst entwickeln. Elektor Verlag, 2012
- Weitere Hinweise zu Online-Quellen und Literatur werden zu Beginn der Veranstaltung veröffentlicht.

**Dozenten:** Habedank, Naroska, Rother**Modulverantwortliche:** Naroska**Aktualisiert:** 06.02.2019

<b>Modul</b>	<b>BWL Grundlagen der Betriebswirtschaft</b>		<b>Credits: 4</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Modul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	1. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	45
<b>Übung</b>	1	15	30
<b>Praktikum</b>			
<b>Arbeitsaufwand in Stunden</b>		45	75
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> keine			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> schriftliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls <ul style="list-style-type: none"> <li>• haben Grundkenntnisse der Betriebswirtschaft;</li> <li>• haben ein Verständnis grundlegender betriebswirtschaftlicher Methoden;</li> <li>• können grundlegende betriebswirtschaftliche Methoden anwenden und dadurch ökonomisch handeln</li> <li>• wissen, wie Entscheidungen in Unternehmen zustande kommen und können dadurch zielgerichtet daran mitwirken</li> </ul>			
<b>Inhalte:</b> Betriebliche Produktionsfaktoren, Externes und Internes Rechnungswesen, Finanzierung und Investitionsrechnung			
<b>Lehrmethoden:</b> Vorlesung, Übung, Selbststudium in der Literatur und im Internet, insbesondere Analyse von Firmeninformationen			
<b>Bezug zu anderen Fächern/Modulen:</b> Die Kenntnisse aus diesem Modul werden im Modul Einführung in das Zivilrecht benötigt			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Thommen/Achleitner: Einführung in die Betriebswirtschaftslehre</li> <li>• Göpel-Gruner: BWL für Informatiker und Ingenieure</li> </ul>			
<b>Dozenten:</b> Göpel-Gruner			
<b>Modulverantwortliche:</b> Göpel-Gruner			
<b>Aktualisiert:</b> 22.05.2018			

<b>Modul</b>	<b>ENG Technisches Englisch</b>		<b>Credits: 3</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Englisch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	1. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	60
<b>Übung</b>			
<b>Praktikum</b>			
<b>Arbeitsaufwand in Stunden</b>		30	60
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Schulkenntnisse in Englisch			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> Testat			
<b>Notensystem:</b>			
<b>Lernziele/Kompetenzen:</b> Die / der Studierende <ul style="list-style-type: none"> <li>• beherrscht das grundlegende Fachvokabular der Informatik,</li> <li>• kann Fachtexte lesen, verstehen und wiedergeben und</li> <li>• kann die erworbenen Kenntnisse in beruflichen Situationen anwenden.</li> </ul>			
<b>Inhalte:</b> Basis-Fachvokabular der Informatik; Lesen und Besprechen von Fachtexten der Informatik; englische Darstellung einfacher Informatik-Inhalte z.B. Erläuterung von Softwarepaketen, Netzwerkkonfigurationen und Systemkonzepten; Präsentation fachlicher Inhalte; Diskussion von Fragestellungen der Informatik; Business English: emails, telephoning, business conversation			
<b>Lehrmethoden:</b> seminaristischer Unterricht mit häuslicher Vor- und Nachbereitung durch die Studierenden, Selbststudium mit der Lernplattform als Hausarbeit			
<b>Bezug zu anderen Fächern/Modulen:</b> In allen weiterführenden Modulen wird die Beherrschung des englischen Fachvokabulars vorausgesetzt sowie die Fähigkeit, Texte in englischer Sprache zu verstehen und fachliche Inhalte in englischer Sprache wiederzugeben; z.B. um englischsprachige Originaldokumentation sowie Fachliteratur zu verstehen und an internationalen Tagungen und Projekten teilzunehmen.			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Oxford English for Information Technology und TechnoPlus (Eurokey)</li> <li>• Lexikon Deutsch-Englisch z.B. Langenscheidt</li> <li>• ausgewählte Fachtexte</li> </ul>			
<b>Dozenten:</b> Hilbrich			
<b>Modulverantwortliche:</b> Hilbrich			
<b>Aktualisiert:</b> 12.06.2018			

<b>Modul</b>	<b>MA2 Mathematik 2</b>		<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	2. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	4	60	30
<b>Übung</b>	2	30	60
<b>Praktikum</b>			
<b>Arbeitsaufwand in Stunden</b>		90	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Kenntnisse und Anwendungskompetenz der Grundlagen mathematischen Schließens und mathematischer Formulierung sowie Grundkenntnisse der linearen Algebra			
<b>Prüfungsvorleistung:</b>			
<b>Prüfungsform:</b> schriftliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Die Studierenden können Methoden und Verfahren der linearen Algebra und <ul style="list-style-type: none"> <li>• Analysis anwenden. Sie sind in der Lage vom Einzelfall zu abstrahieren und geeignete mathematische Methoden nach logischen und algorithmischen Gesichtspunkten auszuwählen und anzuwenden.</li> </ul>			
<b>Inhalte:</b> Lineare Algebra: LR-Zerlegung, Determinante, Eigenwerte, Skalarprodukt, orthogonale Abbildungen, homogene Koordinaten; Analysis: Reelle und komplexe Zahlen, Unendliche Reihen, Potenzreihen, Differentialrechnung, Integralrechnung			
<b>Lehrmethoden:</b> Vorlesung mit Literatur zum Selbststudium; Rechnen von Aufgaben in den Übungen und als Hausübungen			
<b>Bezug zu anderen Fächern/Modulen:</b> Das Modul vermittelt in vielen Modulen des Studiengangs benötigte weitergehende mathematische Kenntnisse zur Analyse und Lösung von Problemstellungen der Informatik.			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• P. Hartmann: Mathematik für Informatiker, Vieweg</li> <li>• Teschl, S. &amp; G.: Mathematik für Informatiker I &amp; II , Springer</li> <li>• P. Stingl: Mathematik für Fachhochschulen Technik und Informatik, Hanser</li> </ul>			
<b>Dozenten:</b> Tipp			
<b>Modulverantwortliche:</b> Tipp			
<b>Aktualisiert:</b> 15.05.2018			

<b>Modul</b>	<b>TEI2 Technische Grundlagen der Informatik 2</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	2. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	3	45	45	
<b>Übung</b>	1	15	15	
<b>Praktikum</b>	1	15	15	
<b>Arbeitsaufwand in Stunden</b>		75	75	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Inhalte des Moduls Technische Grundlagen der Informatik I				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an diesem Modul sind die Studierenden in der Lage, <ul style="list-style-type: none"> <li>• die grundsätzlichen Architekturen von Rechnern,</li> <li>• die Funktionen der einzelnen Rechnerkomponenten sowie ihr Zusammenwirken,</li> <li>• das Programmiermodell eines Rechners,</li> <li>• die Bedeutung und den Umgang mit Speicherhierarchien,</li> <li>• Rechnerstrukturen wie CISC/RISC-Prozessoren, Pipelining, Superskalare und parallele Strukturen zu verstehen und sinnvoll anzuwenden sowie hardwarenahe (oft zeitkritische) Programmierung auf solchen Architekturen durchzuführen.</li> </ul>				
<b>Inhalte:</b> Von-Neumann-Architektur-Modell: <ul style="list-style-type: none"> <li>• –Bussysteme</li> <li>• -Arithmetisch-logische Einheit (ALU)</li> <li>• Register</li> <li>• Arbeitsspeicher</li> <li>• Befehlsformate und Adressrechenwerk</li> <li>• Befehlssatz und Mikroprogrammsteuerwerk</li> <li>• Maschinen- und Assembler-Code</li> <li>• Gegenüberstellung Von-Neumann/Harvard-Architektur</li> <li>•</li> <li>• Speicherhierarchie: <ul style="list-style-type: none"> <li>• Speichertypen</li> <li>• virtueller Speicher</li> <li>• Cache-Organisation</li> </ul> </li> <li>•</li> <li>• CISC/RISC: <ul style="list-style-type: none"> <li>• Adressierungskonzepte</li> <li>• Pipelining</li> <li>• Superskalare/Parallele Architektur</li> </ul> </li> <li>•</li> <li>• Embedded Systems: <ul style="list-style-type: none"> <li>• Vorstellung ausgewählter Mikrocontroller</li> <li>• Programmiermodell</li> <li>• Hardwarenahe Programmierung</li> <li>• Unterprogrammtechnik</li> <li>• Interrupttechnik</li> <li>• Ein-/Ausgabeorganisation</li> <li>• Peripherie- und Timerfunktionen</li> </ul> </li> </ul>				

<b>Lehrmethoden:</b> <ul style="list-style-type: none"><li>• Vorlesung mit Skript und Literatur zum Selbststudium</li><li>• Rechnen von Aufgaben in Hausübungen und Vortrag in den Übungsstunden sowie Nachbereitung im Selbststudium</li><li>• Vorbereitung der Laborarbeit im Selbststudium</li><li>• Lösung hardwarenaher Programmieraufgaben im Labor</li><li>• selbständige Durchführung einer (nicht vorbereiteten) Tagesaufgabe im Labor</li></ul>
<b>Bezug zu anderen Fächern/Modulen:</b> Baut auf dem Modul Technische Grundlagen der Informatik I auf und vermittelt grundlegende Kenntnisse, die in vielen Modulen wie z.B. 'Echtzeitsysteme' benötigt werden.
<b>Literatur:</b> <ul style="list-style-type: none"><li>• Skript zur Vorlesung</li><li>• Tanenbaum: Rechnerarchitektur: Von der digitalen Logik zum Parallelrechner. Pearson Studium München, 2014</li><li>• Wiegmann: Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller: C-Programmierung für Embedded-Systeme. VDE VERLAG GmbH, 2017</li><li>• Wüst: Mikroprozessortechnik: Grundlagen, Architekturen, Schaltungstechnik und Betrieb von Mikroprozessoren und Mikrocontrollern. Vieweg+Teubner Verlag, 2010</li><li>• Aufgrund der Entwicklungsgeschwindigkeit des Themengebiets und der Fluktuation bei Publikationen werden weitere Hinweise zu Online-Quellen und Literatur zu Beginn der Veranstaltung veröffentlicht.</li></ul>
<b>Dozenten:</b> Habedank, Naroska, Brandt
<b>Modulverantwortliche:</b> Habedank
<b>Aktualisiert:</b> 22.05.2018

<b>Modul</b>	<b>ALD Algorithmen und Datenstrukturen</b>		<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Modul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	2. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	45
<b>Übung</b>	2	30	45
<b>Praktikum</b>			
<b>Arbeitsaufwand in Stunden</b>		60	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Grundlegende Programmierkenntnisse und Kenntnisse von grundlegenden mathematischen Verfahren			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> schriftliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Absolventen des Moduls <ul style="list-style-type: none"> <li>• können konkrete Problemstellungen abstrahieren</li> <li>• kennen Lösungsmuster für diese Problemstellungen</li> <li>• können Lösungsmuster in effiziente Algorithmen umsetzen</li> <li>• kennen geeignete Datenstrukturen für die Algorithmen</li> <li>• können eine Laufzeitanalyse der Algorithmen durchführen</li> </ul>			
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Datenstrukturen: Stack, Liste, Bäume, Hash-Tabelle, Heap</li> <li>• Komplexität und asymptotischen Aufwandsabschätzung, Landau-Symbole</li> <li>• Suchverfahren: Binärsuche, Interpolationssuche, Suchbäume</li> <li>• Sortierverfahren: Quicksort, Mergesort, Heapsort, Radixsort, untere Schranke</li> <li>• Graphalgorithmen: Breitensuche, Tiefensuche, Minimaler Spannbaum, kürzeste Wege, TSP</li> </ul>			
<b>Lehrmethoden:</b> Vorlesung und Literatur zum Selbststudium, Lösen von Aufgaben zu Hause und besprechen der Lösungen in der Übung an der Tafel und am Rechner			
<b>Bezug zu anderen Fächern/Modulen:</b> Baut auf dem Modul Einführung in die Programmierung auf. Grundlegende Kenntnisse über den Entwurf von Algorithmen und Datenstrukturen werden in fast allen weiterführenden Modulen benötigt.			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Heun: Grundlegende Algorithmen. Vieweg</li> <li>• Sedgewick: Algorithmen. Pearson Studium/Addison-Wesley</li> <li>• Ottmann, Widmayer: Algorithmen und Datenstrukturen.</li> <li>• Cormen, Leiserson, Rivest: Introduction to Algorithms. MIT Press</li> </ul>			
<b>Dozenten:</b> Rethmann, Ueberholz			
<b>Modulverantwortliche:</b> Ueberholz			
<b>Aktualisiert:</b> 23.05.2018			

<b>Modul</b>	<b>BSY Betriebssysteme</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Modul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	2. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	45	
<b>Übung</b>	1	15	15	
<b>Praktikum</b>	1	15	15	
	<b>Arbeitsaufwand in Stunden</b>	60	75	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Programmierkenntnisse in C und Kenntnisse in Rechnerarchitekturen				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls können <ul style="list-style-type: none"> <li>• den Aufbau eines modernen Betriebssystems beschreiben,</li> <li>• systembezogene Abstraktionen, wie sie von Betriebssystemen bereitgestellt werden, verstehen und wieder erkennen,</li> <li>• einschlägige Algorithmen und Strategien zur effizienten Verwaltung und fairen Vergabe der verschiedenen Betriebsmittel skizzieren,</li> <li>• die Umsetzung einiger Mechanismen in aktuellen Betriebssystemen beschreiben,</li> <li>• Probleme bei der Interprozesskommunikation identifizieren und Lösungsmöglichkeiten vorschlagen und</li> <li>• Programme unter Nutzung der UNIX-Programmierschnittstelle (API) entwickeln und Beispielimplementierungen bei Verwendung der Windows-Programmierschnittstelle verstehen</li> </ul>				
<b>Inhalte:</b> Architekturen von Betriebssystemen; Benutzerschnittstelle in UNIX/Linux; Multiprogramming; Prozesse, Threads, Scheduling-Strategien; Speicherverwaltung: Speicherpartitionierung, virtueller Speicher; Techniken der Synchronisation und Interprozesskommunikation; Geräte- und Dateiverwaltung; Beispielimplementierung für Prozesse, Threads und Synchronisationsmechanismen unter Linux und Windows				
<b>Lehrmethoden:</b> Vorlesung mit Foliensammlung, Skript, Literatur und Beispielprogramme zum Selbststudium, Lösen von Aufgaben in den Übungsstunden, Schreiben von C-Programmen unter Nutzung des Raspberry Pi im Praktikum, theoretische Vorbereitung des Praktikums im Selbststudium mit Nutzung der Lernplattform moodle.				
<b>Bezug zu anderen Fächern/Modulen:</b> Grundlegende Kenntnisse von Betriebssystemen werden in den Modulen Verteilte Systeme und Echtzeitsysteme vorausgesetzt. Sie sind weiterhin für alle Module wichtig, die sich mit Software-Entwicklung beschäftigen sowie für die Module DNÜ und IT-Sicherheit.				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• A. S. Tanenbaum: Moderne Betriebssysteme; Pearson Studium, 2016.</li> <li>• E. Glantz: Betriebssysteme, dpunkt.verlag, 2010</li> <li>• R. Bause: Betriebssysteme, Grundlagen und Konzepte, Springer Vieweg, 2017</li> <li>• P. Mandl: Grundkurs Betriebssysteme, Springer Vieweg, 2014</li> </ul>				
<b>Dozenten:</b> Pohle-Fröhlich				
<b>Modulverantwortliche:</b> Pohle-Fröhlich				
<b>Aktualisiert:</b> 16.05.2018				

<b>Modul</b>	<b>OOA Objektorientierte Anwendungsentwicklung</b>			<b>Credits: 7</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	2. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	3	45	60	
<b>Übung</b>	2	30	15	
<b>Praktikum</b>	2	30	30	
<b>Arbeitsaufwand in Stunden</b>		105	105	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Programmierkenntnisse in C				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls <ul style="list-style-type: none"> <li>• können Programme mit objektorientierten Methoden und Techniken entwerfen und programmieren</li> <li>• können Anforderungen in einen effizienten Algorithmus und eine Datenstruktur umsetzen</li> <li>• sind vertraut mit Lösungsmustern</li> <li>• erkennen Inkonsistenzen und können mit unklaren Anforderungen umgehen</li> <li>• besitzen die Fähigkeit zur Abstraktion</li> <li>• beherrschen die gängigen Programmiersprachen C und C++</li> <li>• können sich in vorhandene Programme einarbeiten und vorhandene Programmelemente nutzen</li> </ul>				
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Grundlagen des objektorientierten Anwendungsentwurfs mit UML</li> <li>• Grundlagen der objektorientierten und generischen Programmierung in C++</li> <li>• Anwenden der Standard Template Library, von Entwurfsmustern und Refactoring-Methoden</li> </ul>				
<b>Lehrmethoden:</b> <ul style="list-style-type: none"> <li>• Vorlesung mit Diskussion der Fachinhalte und der Probleme.</li> <li>• Bearbeiten von Übungsaufgaben am Rechner mit Diskussion der Lösungen.</li> <li>• Im Praktikum sollen die Studierenden in kleinen Teams Programme entwerfen und implementieren. Die Methoden sollen zu Hause vertieft werden.</li> </ul>				
<b>Bezug zu anderen Fächern/Modulen:</b> Baut auf dem Modul Einführung in die Programmierung (EPR) auf und dient als Vorbereitung für viele Module in höheren Semestern wie "Interaktive Systeme", "Graphische DV" und "Bildbearbeitung", "Datenbanksysteme" und "Verteilte Systeme".				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• M. Schader, S. Kuhlins: Programmieren in C++. Springer.</li> <li>• S. Kuhlins, M. Schader: Die C++ Standardbibliothek. Springer.</li> <li>• E. Freeman, E. Freeman: Entwurfsmuster von Kopf bis Fuß. O'Reilly.</li> <li>• E. Gamma, R. Helm, R. Johnson, J. Vlissides: Entwurfsmuster. Addison-Wesley.</li> <li>• M. Fowler: Refactoring. Addison-Wesley</li> <li>• H. Herold, M. Klar, S. Klar: C++, UML und Design Patterns. Addison-Wesley</li> <li>• H. Balzert: Lehrbuch der Objektmodellierung. Spektrum.</li> <li>• B. Oestereich: Objektorientierte Softwareentwicklung. Oldenbourg.</li> </ul>				
<b>Dozenten:</b> Davids, Rethmann				
<b>Modulverantwortliche:</b> Rethmann				
<b>Aktualisiert:</b> 10.05.2018				

<b>Modul</b>	<b>MAR Grundlagen des Marketing</b>			<b>Credits: 4</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Modul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	2. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	45	
<b>Übung</b>	1	15	30	
<b>Praktikum</b>				
<b>Arbeitsaufwand in Stunden</b>		45	75	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> keine				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Modul <ul style="list-style-type: none"> <li>• kennen Grundlagen des Marketings</li> <li>• haben ein Verständnis für grundlegende Marketing-Strategien und Instrumente unter besonderer Berücksichtigung des E-Marketing und können diese anwenden</li> </ul>				
<b>Inhalte:</b> Einführung in das Marketing, Business-to-Business-Marketing, Grundlagen des E-Marketing, Marketingstrategien, Marketinginstrumente, Marktanalyse				
<b>Lehrmethoden:</b> Vorlesung mit Literatur zum Selbststudium; selbstständiges Bearbeiten von Aufgaben und Fallstudien in den Übungen				
<b>Bezug zu anderen Fächern/Modulen:</b> Die Kenntnisse aus diesem Modul werden im Master Informatik benötigt				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Kotler: Grundlagen des Marketing</li> <li>• Vorlesungsskript</li> </ul>				
<b>Dozenten:</b> Göpel-Gruner				
<b>Modulverantwortliche:</b> Göpel-Gruner				
<b>Aktualisiert:</b> 22.05.2018				

<b>Modul</b>	<b>STA Statistik</b>			<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	3. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	4	60	60	
<b>Übung</b>	2	30	30	
<b>Praktikum</b>	0	0	0	
<b>Arbeitsaufwand in Stunden</b>		90	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Mathematische Kenntnisse in der linearen Algebra (Vektoren, Skalarprodukt) und Analysis (Differentialrechnung, Integralrechnung)				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Modul <ul style="list-style-type: none"> <li>• -können zufällige Vorgänge mit statistischen Modellen beschreiben</li> <li>• -können Wahrscheinlichkeiten berechnen</li> <li>• -kennen statistische Kenngrößen und können sie berechnen</li> <li>• -können Beobachtungen auf ihre statistische Signifikanz testen</li> <li>• -verstehen das Prinzip statistischer Tests und können statistische Auswertungen anderer hinterfragen</li> </ul>				
<b>Inhalte:</b> Statistische Datenbeschreibung: Merkmale und Skalen, Häufigkeiten und deren Darstellung, Kenngrößen von Verteilungen (Mittelwert, Streuung, Quantile), Korrelation, Least-Squares-Fit <ul style="list-style-type: none"> <li>• Wahrscheinlichkeitsrechnung: Zufallsexperimente, Zufallsereignisse, Wahrscheinlichkeit, Kombinatorik, bedingte Wahrscheinlichkeit, Zufallsgrößen, stetige und diskrete Wahrscheinlichkeitsv</li> <li>• erteilungen, Gesetze der großen Zahlen, Summenverteilungen und zentraler Grenzwertsatz</li> <li>Statistisches Schätzen: Schätzen von Parametern, Erwartungstreue und Konsistenz, Testen von</li> <li>• Hypothesen</li> </ul>				
<b>Lehrmethoden:</b> Vorlesung mit Literatur zum Selbststudium; selbstständiges Rechnen von Aufgaben in den Übungen				
<b>Bezug zu anderen Fächern/Modulen:</b> Das Modul vermittelt in vielen Modulen des Studiengangs benötigte statistische Kenntnisse				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• -Fahrmeir, Künstler, Pigeot, Tutz: Statistik. Springer 2004</li> <li>• -Knuth: The Art of Computer Programming, Volume 2, Addison-</li> <li>• Wesley 1981</li> <li>• -Krengel: Einführung in die Wahrscheinlichkeitstheorie und Statistik. Vieweg</li> </ul>				
<b>Dozenten:</b> Dalitz				
<b>Modulverantwortliche:</b> Dalitz				
<b>Aktualisiert:</b> 11.05.2018				

<b>Modul</b>	<b>GRA Graphische DV und Bildverarbeitung</b>		<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	3. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	3	45	30
<b>Übung</b>	2	30	15
<b>Praktikum</b>	1	15	15
<b>Arbeitsaufwand in Stunden</b>		90	60
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Programmierkenntnisse in C			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> schriftliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an diesem Modul ist der Studierende in der Lage, grundlegende Algorithmen auf den Gebieten der 2D und 3D Computergraphik sowie der Bildverarbeitung zu erklären, Three.js für Graphik und Interaktion zu nutzen, Problemstellungen für die Einsatzmöglichkeit der digitalen Bildverarbeitung zu bewerten und Problemlösungen für praktische Bildverarbeitungsfragestellungen mit Hilfe von OpenCV zu erstellen			
<b>Inhalte:</b> Computergraphik: Rasterung von Linien und Kreisen, Anti <ul style="list-style-type: none"> <li>• -Aliasing, Füllalgorithmen, Koordinatensysteme und Transformationen, Geometrirepräsentation, Hidden Surface Removal, Beleuchtungsberechnung, Shading, Textur; Bildverarbeitung: Bildaufnahme, Fouriertransformation,</li> <li>• Bildrestauration, Bildverbesserung, Segmentierung, morphologische Operationen</li> </ul>			
<b>Lehrmethoden:</b> Vorlesung mit Foliensammlung, Beispielprogrammen und Literatur zum Selbststudium, Lösen von theoretischen Aufgaben in den Übungsstunden, Schreiben von OpenCV- und Three.js- <ul style="list-style-type: none"> <li>• Programmen in den Übungen und im Praktikum, theoretische Vorbereitung des Praktikums im Selbststudium mit Nutzung der Lernplattform moodle</li> </ul>			
<b>Bezug zu anderen Fächern/Modulen:</b> Das vorliegende Modul benötigt die Mathematik des ersten Studienjahres und Kenntnisse aus der Veranstaltung Algorithmen und Datenstrukturen. Kenntnisse aus der Veranstaltung Einführung in die Programmierung werden genutzt und gefestigt. <ul style="list-style-type: none"> <li>•</li> </ul>			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• -Nischwitz, M. Fischer, P. Haberäcker, G. Socher: Computergraphik und Bildverarbeitung, Teil 1: Computergraphik, Teil 2: Bildverarbeitung, Springer Verlag, 2012.</li> <li>• -K.D. Tönnies: Grundlagen der Bildverarbeitung, Person Studium, 2005.</li> <li>• -W. Burger, M. Burge: Digitale Bildverarbeitung, Springer Verlag, 2015.</li> </ul>			
<b>Dozenten:</b> Pohle-Fröhlich			
<b>Modulverantwortliche:</b> Pohle-Fröhlich			
<b>Aktualisiert:</b> 16.05.2018			

<b>Modul</b>	<b>WEB Web-Engineering</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	3. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	30	
<b>Übung</b>	1	15	30	
<b>Praktikum</b>	1	15	30	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Der Studierende beherrscht die imperative Programmierung (Programmiersprache C) und die objektorientierte Anwendungsentwicklung (Programmiersprache C++)				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen des Moduls sind vertraut mit Lösungsmustern, erkennen und formulieren funktionale und nichtfunktionale Anforderungen, können abstrahieren, modellieren anwendungsgerechte und ergonomische Mensch-Maschine-Schnittstellen, kennen Entwicklungsumgebungen, sind kontaktfähig und arbeiten in Gruppen, können einem Text wichtige Inhalte entnehmen, diese strukturieren und wiedergeben, nutzen verschiedene Medien zur Informationsbeschaffung, können vorhandene Missverständnisse zwischen Gesprächspartnern erkennen und abbauen, führen ein Ziel- und Zeitmanagement aus				
<b>Inhalte:</b> Webbasierte Technologien : technische Grundlagen (Adressierung, Client-Server-Architekturen, Software-Agenten, service-orientierte Architekturen), verbreitete Techniken (z.B. Markup Sprachen, Standardprotokolle) <ul style="list-style-type: none"> <li>• Webbasierte Systeme : Architektur, Anwendungsbereiche und -formen, Entwicklungsmethoden und -werkzeuge, Analyse web-basierter Systeme: Qualitätsanalysen, Performance-Analysen, Test web-basierter Systeme, Web-basierte Systeme als wesentlicher Bestandteil kollaborativer Software</li> </ul>				
<b>Lehrmethoden:</b> Vorlesung mit Skript und Literatur zum Selbststudium; Bearbeitung von Aufgaben in den Übungsstunden; Bearbeitung von projektartigen Aufgabenstellungen <ul style="list-style-type: none"> <li>• zur Entwicklung webbasierter Anwendungen in kleinen Teams mit theoretischer Vorbereitung, praktischer Einführung in die Arbeitsumgebung, Entwurf, Implementierung und Test</li> <li>•</li> </ul>				
<b>Bezug zu anderen Fächern/Modulen:</b> Das Modul gehört zu den Modulen des Themengebiets "Software-Engineering". Es ist eine Fortsetzung des Moduls "Objektorientierte Anwendungsentwicklung" im Hinblick auf den Themenbereich World-Wide-Web. Die erarbeiteten Kenntnisse und Fähigkeiten werden weiter genutzt im Modul "Interaktive Systeme", Aspekte des Web-Engineering werden im Modul "Software-Engineering" ausgebaut und vertieft.				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Philip Ackermann: Professionell entwickeln mit JavaScript / Design, Patterns, Praxistipps, Rheinwerk Computing</li> <li>• Dane Cameron: HTML5, JavaScript und jQuery / Der Crashkurs für Softwareentwickler, dpunkt.verlag</li> <li>• Nicholas C. Zakas: JavaScript objektorientiert, dpunkt.verlag</li> <li>• Castro, Elizabeth / Hyslop, Bruce: Praxiskurs HTML5 &amp; CSS3 / Professionelle Webseiten von Anfang an, dpunkt.verlag</li> <li>• Peter Müller: Flexible Boxes / Eine Einführung in moderne Websites, Rheinwerk Verlag</li> <li>• Weitere aktuelle Literaturrempfehlungen werden zu Beginn der Vorlesung zur Verfügung gestellt.</li> </ul>				
<b>Dozenten:</b> Beims				
<b>Modulverantwortliche:</b> Beims				
<b>Aktualisiert:</b> 23.05.2018				

<b>Modul</b>	<b>THI Theoretische Informatik</b>		<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	3. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	45
<b>Übung</b>	2	30	45
<b>Praktikum</b>	0	0	0
<b>Arbeitsaufwand in Stunden</b>		60	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Grundlegende mathematische Verfahren, Grundkenntnisse in Algorithmen			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> schriftliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls können Probleme abstrahieren und klassifizieren, können einfache Probleme mit Automaten und formalen Sprachen beschreiben, finden geeignete mathematische Abstraktionen für konkrete Problemsituationen und können Laufzeitanalysen von Problemen durchführen und Probleme in Klassen einteilen			
<b>Inhalte:</b> Die Vorlesung gibt eine Einführung in die Gebiete "Berechenbarkeit", "Komplexitätstheorie", "Formale Sprachen" und "Automatentheorie". Es werden folgende Themen behandelt: <ul style="list-style-type: none"> <li>• Berechenbarkeitsbegriff, Turingmaschinen, unlösbare Probleme Laufzeitanalyse von Algorithmen, Komplexitätsklassen P und NP, NP-Vollständigkeit Grammatiken und Chomsky Hierarchie, endliche und Keller Automaten, Reguläre Ausdrücke, Pumping Lemma</li> </ul>			
<b>Lehrmethoden:</b> Vorlesung und selbständiges Bearbeiten von Übungsaufgaben; ergänzende Literatur zum Selbststudium			
<b>Bezug zu anderen Fächern/Modulen:</b> <ul style="list-style-type: none"> <li>• Die Abschnitte "Berechenbarkeit" und "Komplexität" vermitteln eine allgemeinere Sicht auf die im Modul "Algorithmen und Datenstrukturen" vermittelten Algorithmen.</li> <li>• Die "Automatentheorie" verallgemeinert und vertieft, die im Modul "Technische Grundlagen der Informatik I" behandelten endlichen Automaten.</li> <li>• "Formale Sprachen" schaffen die Voraussetzung für Wahlmodule, in denen das Parsen von Text eine Rolle spielt (z.B. Compilerbau, XML).</li> </ul>			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Sipser: Introduction to the Theory of Computation. PSW Publishing 1997</li> <li>• Asteroth, Baier: Theoretische Informatik. Pearson Studium, 2002</li> <li>• Vossen und Witt: Grundkurs Theoretische Informatik. Vieweg Verlag, 2006</li> </ul>			
<b>Dozenten:</b> Dalitz, Ueberholz			
<b>Modulverantwortliche:</b> Ueberholz			
<b>Aktualisiert:</b> 23.05.2018			

<b>Modul</b>	<b>DNÜ Datennetze und Datenübertragung</b>		<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	3. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	3	45	60
<b>Übung</b>	2	30	15
<b>Praktikum</b>	1	15	15
	<b>Arbeitsaufwand in Stunden</b>	90	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Die Studierenden verfügen über grundlegende Kenntnisse über den Aufbau und Funktion von Rechner- und Betriebssystemen sowie Algorithmen und Datenstrukturen.			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> schriftliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage: <ul style="list-style-type: none"> <li>• Struktur, Komponenten und Funktion des Internets zu erklären;</li> <li>• Netzwerk-Protokolle anhand von Schichtenmodellen einzuordnen und zu erklären;</li> <li>• grundlegende Anforderungen an Netzwerkstrukturen zu verstehen und geeignete Adressierungsschemata zu entwickeln;</li> <li>• Routing-Protokolle zu verstehen, zu bewerten und einzusetzen;</li> <li>• kleinere Unternehmens-LANs zu verstehen und zu entwerfen;</li> <li>• typische Fehlersituationen in Netzwerken zu analysieren und zu beheben;</li> </ul>			
<b>Inhalte:</b> Einsatz von und Anforderungen an Datennetze; Netzkomponenten, Übertragungsmedien; Netztopologien; Protokollhierarchie; Ausgewählte Protokolle der Applikationsschicht, TCP/IP-Protokollfamilie, IPv4/IPv6-Adressierung; Grundlagen des Routing; Routing-Protokolle; Grundlagen des Switching/LAN-Design/VLANS; IP-Services (DHCP/NAT); Fehlersuche;			
<b>Lehrmethoden:</b> Lehrvortrag in geraffter Form, Foliensammlung und Literatur zum Selbststudium. angeleitetes Lösen von Aufgaben, Versuche mit Netzsimulator PacketTracer und an experimentellen Datennetzen, theoretische Vorbereitung der Versuche und Protokollausarbeitung als Hausarbeit. Vor- und Nachbereitung aller Veranstaltungen und Klausurvorbereitung mit Online-Curriculum.			
<b>Bezug zu anderen Fächern/Modulen:</b> Das vorliegende Modul ist Grundlage für die Module "Datennetzmanagement" und "IT-Sicherheit". Zusammen mit dem Modul "Datennetzmanagement" qualifiziert es die Studierenden für den Erwerb des Industriezertifikats CCNA.			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Kurose, Ross: Computernetzwerke: Ein Top-Down-Ansatz. 6. Auflage, Pearson Studium, 2014</li> <li>• Tanenbaum: Computernetzwerke. 5. Auflage, Pearson Studium, 2012</li> <li>• Odom, Wendell: CCNA Routing and Switching ICND2 200-105 Official Cert Guide, Cisco Press, 2017</li> </ul>			
<b>Dozenten:</b> Weidenhaupt			
<b>Modulverantwortliche:</b> Weidenhaupt			
<b>Aktualisiert:</b> 18.05.2018			

<b>Modul</b>	<b>VSY Verteilte Systeme</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	3. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	45	
<b>Übung</b>	2	30	45	
<b>Praktikum</b>	0	0	0	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> - grundlegende Programmierkenntnisse in C/C++ - grundlegende Kenntnisse in Rechnerarchitekturen und Netzwerktechniken - gute Betriebssystem-Kenntnisse wie Thread-Programmierung, - Scheduling- und Auslagerungsverfahren				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls <ul style="list-style-type: none"> <li>• können verteilte Rechnerstrukturen entwerfen und bewerten</li> <li>• können Client-Server-Strukturen konzipieren und implementieren</li> <li>• können neue Problemstellungen für verteilte Anwendungen analysieren und implementieren</li> <li>• sind sich den vielfältigen Sicherheitsproblemen bewusst und</li> <li>• kennen Techniken und Verfahren für die Sicherung von verteilten IT-Systemen</li> </ul>				
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Verteilte Systemarchitekturen (Beispiel: Peer-to-Peer Netze, Client/Server-Systeme)</li> <li>• Techniken zur Synchronisation und Kommunikation</li> <li>• Nachrichtenorientierte Kommunikation (Beispiel: TCP-IP Sockets)</li> <li>• Auftragsorientierte Kommunikation (Beispiel: Sun RPC)</li> <li>• Objektbasierte Architekturen (Beispiele: Corba, Java RMI)</li> <li>• Webbasierte Systeme (Beispiel: Web-Services)</li> <li>• Dienste (Beispiele: Namens-, Verzeichnis-, Datei-, Zeit- und Transaktionsdienst)</li> <li>• Verteilte Algorithmen (Beispiele: Wahl- und Konsensalgorithmen)</li> </ul>				
<b>Lehrmethoden:</b> Vorlesungen und praktische Übungen am Rechner, in denen das Gewicht auf der Erstellung von Client-Server Systemen liegt. Dazu steht den Studierenden eine Reihe von PCs zur Verfügung, mit denen Sie verteilte Systeme von Grund auf aufbauen sollen. In der Übung soll in kleinen Teams an einem größeren Projekt gearbeitet werden, an dem die Studierenden lernen, ein größeres IT-System zu entwerfen, zu implementieren und zu testen				
<b>Bezug zu anderen Fächern/Modulen:</b> Die Veranstaltung setzt auf diversen Veranstaltungen der ersten beiden Semester auf und ist grundlegend für moderne IT-Systeme.				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• G. Bengel: Verteilte Systeme. Vieweg Verlag</li> <li>• Tanenbaum, M. van Steen: Verteilte Systeme. Pearson Studium</li> <li>• J. Dollimore, T. Kindberg, G. Coulouris: Verteilte Systeme. Pearson Studium</li> <li>• U. Hammerschall: Verteilte Systeme und Anwendungen. Pearson Studium</li> </ul>				
<b>Dozenten:</b> Rethmann, Ueberholz				
<b>Modulverantwortliche:</b> Rethmann				
<b>Aktualisiert:</b> 17.05.2018				

<b>Modul</b>	<b>DBS Datenbanksysteme</b>			<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	4. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	3	45	45	
<b>Übung</b>	2	30	15	
<b>Praktikum</b>	1	15	30	
<b>Arbeitsaufwand in Stunden</b>		90	90	

<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben
<b>Vorkenntnisse:</b> Mathematik: Mengen, Relationen, Kombinatorik, Definition von Graphen Programmieren: solide Kenntnisse der C und C++ Programmierung Algorithmen: Turing-Berechenbarkeit, Laufzeitkomplexität von Such- und Sortieralgorithmen in O-Notation
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben
<b>Prüfungsform:</b> schriftliche benotete Prüfung
<b>Notensystem:</b> deutsche Notenskala 1-5
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls <ul style="list-style-type: none"> <li>• können SQL-basierte Datenbanksysteme bedienen und programmieren</li> <li>• können eine Praxisanforderung in ein relationales Datenmodell umzusetzen</li> <li>• können die Qualität eines Datenmodells beurteilen</li> <li>• kennen die Probleme der Nebenläufigkeit und können Lösungsstrategien benennen</li> </ul>
<b>Inhalte:</b> Die Veranstaltung befasst sich schwerpunktmäßig mit den heute dominierenden "relationalen Datenbanken" und behandelt die Fragen der Datenbankanwendung, Datenmodellierung und einzelne Teilbereiche der Datenbankimplementierung: (a) Relationales Modell, Relationale Algebra, Normalformen, Entity-Relationship Modell (b) SQL, Clientseitige und Serverseitige Programmierung, Datenbanktuning (c) Transaktionen: ACID-Anforderungen, Concurrency-Control Verfahren
<b>Lehrmethoden:</b> Vorlesung mit theoretischen Übungen, selbständige Umsetzung praxisnaher Kleinprojekte am Rechner im Praktikum
<b>Bezug zu anderen Fächern/Modulen:</b> Wendet in den Modulen "Mathematik 1", "Einführung in die Programmierung" und "Objektorientierte Anwendungsentwicklung" erworbenen Kenntnisse an.
<b>Literatur:</b> Elmasri, Navathe: Grundlagen von Datenbanksystemen. Pearson Studium 2002 <ul style="list-style-type: none"> <li>• Eisentraut: PostgreSQL - Das offizielle Handbuch. Mitp 2003</li> <li>• Ein Vorlesungsskript ist von der Webseite des Dozenten erhältlich</li> </ul>
<b>Dozenten:</b> Dalitz, Rethmann, Weidenhaupt
<b>Modulverantwortliche:</b> Dalitz
<b>Aktualisiert:</b> 11.05.2018

<b>Modul</b>	<b>DNM Datennetzmanagement</b>			<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	4. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	3	45	45	
<b>Übung</b>	2	30	15	
<b>Praktikum</b>	1	15	30	
<b>Arbeitsaufwand in Stunden</b>		90	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Grundlagen Datennetze (Standards und ISO-Modell, Topologien), Netzkomponenten: Router und Switches, TCP/IP-Protokolle, IP-Adressierung; Routing und Router-Programmierung				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Vermittlung anwendungsorientierter Kenntnisse und Fähigkeiten in der Netzadministration, der Design- und Realisierungskompetenz für den Entwurf, die Entwicklung und den Einsatz von Netzwerken, technologischer Kompetenzen zu deren Betrieb und einer umfassenden Methodenkompetenz zur Entwicklung von Problemlösungskonzepten. Absolventen dieses Moduls sind in der Lage, <ul style="list-style-type: none"> <li>• Konzepte und Funktionsweisen von Rechnernetzen zu verstehen</li> <li>• Unternehmensnetze mit der entsprechenden Hard- und Software zu konstruieren, die die Anforderungen an das System vollständig erfüllen</li> <li>• ein Bewusstsein für die vielfältigen Sicherheitsprobleme zu entwickeln, sowie Techniken und Verfahren zur Sicherung von Unternehmensnetzen einzusetzen</li> <li>• Probleme im Netzwerk zu analysieren, zu lokalisieren und eine Problemlösung zu erstellen.</li> </ul>				
<b>Inhalte:</b> Adressmanagement für IPv4 und IPv6; NAT; Design, Aufbau und Betrieb redundanter LANs; WLAN-Technologien, ihr Einsatz und WLAN-Sicherheit; Dynamisches Routing in Unternehmensnetzen mit EIGRP und OSPF; Sichere Anbindung von Unternehmensnetze an das Internet (Techniken, PPP, VPN, PPPoE, eBGP, Access Listen); Netzwerkmanagement, insb. mit SNMP; Sicherheit in Netzwerken; QoS-Sicherung der Übertragungsqualität; Trends in der Netzwerkentwicklung (SDN, IoT, Cloud/Virtialisierung)				
<b>Lehrmethoden:</b> Vorlesung, Übungen, Praktikum im Laborraum mit schriftlicher Ausarbeitung, zusätzliche praktische Übungen im Lernmodul Packet Tracer, Online-Tests, Vor- und Nachbereitung aller Veranstaltungen und Klausurvorbereitung mit Online-Curriculum				
<b>Bezug zu anderen Fächern/Modulen:</b> Zusammen mit den Inhalten des Moduls "Datennetze und Datenübertragung" sind die Kenntnisse und Fähigkeiten zu erlangen, die zum Erwerb des Industriezertifikats CCNA für Netzwerkspezialisten erforderlich sind.				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• A.S. Tanenbaum: Computer Networks , Pearson New International Edition, Juli 2013, Prentice Hall International, ISBN 978-1292024226</li> <li>• W. Riggert: Rechnernetze, 5. Auflage Juni 2014, Hanser Fachbuch, ISBN 978-3446442047</li> <li>• A. Badach, E. Hoffmann: Technik der IP-Netze. Internet-Kommunikation in Theorie und Einsatz , 3. Aufl. November 2014, Hanser Fachbuch, ISBN 978-3446439764</li> <li>• R&amp;S Essentials v6 Companion Guide, published Dec 2016 by Cisco Press, ISBN 978-1-58713-428-9</li> <li>• Scaling Networks v6 Companion Guide, published Dec 2016 by Cisco Press, ISBN 978-1-58713-434-0</li> <li>• Connecting Networks v6 Companion Guide; published Dec 2016 by Cisco Press, ISBN 978-1-58713-432-6</li> </ul>				
<b>Dozenten:</b> Meuser				
<b>Modulverantwortliche:</b> Meuser				
<b>Aktualisiert:</b> 20.05.2018				

<b>Modul</b>	<b>IAS Interaktive Systeme</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Modul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	4. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	30	
<b>Übung</b>	1	15	30	
<b>Praktikum</b>	1	15	30	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Der Studierende beherrscht die imperative Programmierung (Programmiersprache C) und die objektorientierte Anwendungsentwicklung (Programmiersprache C++) und kann webbasierte Anwendungssysteme konzipieren und implementieren (HTML, CSS, Programmiersprachen Javascript und Python).				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Absolventen des Moduls <ul style="list-style-type: none"> <li>• sind vertraut mit Lösungsmustern</li> <li>• erkennen und formulieren funktionale und nichtfunktionale Anforderungen</li> <li>• definieren Schnittstellen so, dass die Systeme wartbar, erweiterbar und zuverlässig sind</li> <li>• können abstrahieren</li> <li>• modellieren anwendungsgerechte und ergonomische Mensch-Maschine-Schnittstellen</li> <li>• sind kontaktfähig und arbeiten in Gruppen</li> <li>• können einem Text wichtige Inhalte entnehmen, diese strukturieren und wiedergeben</li> <li>• nutzen verschiedene Medien zur Informationsbeschaffung</li> <li>• können vorhandene Missverständnisse zwischen Gesprächspartnern erkennen und abbauen</li> <li>• führen ein Ziel- und Zeitmanagement aus</li> </ul>				
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Software-Ergonomie : Grundlagen, Normen, Barrierefreiheit, Konsequenzen für die Anwendungsentwicklung</li> <li>• Architektur ereignisgesteuerter Systeme</li> <li>• Entwurfsmuster für ereignisgesteuerte Systeme</li> <li>• Implementierung portabler interaktiver System</li> </ul>				
<b>Lehrmethoden:</b> Vorlesung mit Skript und Literatur zum Selbststudium; Bearbeitung von Aufgaben in den Übungsstunden; Bearbeitung von projektartigen Aufgabenstellungen zur Konzeption von interaktiven Anwendungen in kleinen Teams.				
<b>Bezug zu anderen Fächern/Modulen:</b> Das Modul gehört zu den Modulen des Themengebiets "Software-Engineering". Es ist eine Fortsetzung der Module "Objektorientierte Anwendungsentwicklung" und "Web-Engineering" und ist Grundlage des Moduls "Software-Engineering"				
<b>Literatur:</b> B. Shneiderman: Designing the User Interface, Pearson/Addison-Wesley <ul style="list-style-type: none"> <li>• Markus Dahm: Grundlagen der Mensch-Computer-Interaktion, Pearson Studium</li> <li>• A. Butz, A. Krüger: Mensch-Maschine-Interaktion, De Gruyter / Oldenbourg</li> <li>• J. Jacobsen, L. Meyr: Praxisbuch Usability und UX, Rheinwerk-Verlag</li> <li>• Weitere aktuelle Literaturempfehlungen werden zu Beginn der Vorlesung bereitgestellt</li> </ul>				
<b>Dozenten:</b> Beims				
<b>Modulverantwortliche:</b> Beims				
<b>Aktualisiert:</b> 23.05.2018				

<b>Modul</b>	<b>WPV Numerik</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Wahlpflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	4. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	30	
<b>Übung</b>	2	30	60	
<b>Praktikum</b>	0	0	0	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Grundlegende mathematische Verfahren. z.B. aus Mathematik 1 und Mathematik 2				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> mündliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Der Schwerpunkt dieses Moduls liegt darin, das logische Denken zu fördern. Algorithmen und mathematische Kompetenzen sowie Realisierungskompetenzen für numerische Problemstellungen zu stärken. Dazu sollen geeignete mathematische Modell für konkrete Problemsituationen theoretisch entwickelt werden, anschließend auf Rechnern In einer Programmiersprache realisiert werden und die Grenzen der numerischen Behandlung untersucht werden.				
<b>Inhalte:</b> In der Veranstaltung werden folgende Themen behandelt: <ul style="list-style-type: none"> <li>• Rechenarithmetik und Rundungsfehler</li> <li>• Lineare Gleichungssysteme</li> <li>• Lineare Ausgleichsrechnung</li> <li>• Interpolation</li> <li>• Numerische Differentiation und Integration</li> <li>• Das Nullstellenproblem</li> </ul>				
<b>Lehrmethoden:</b> Vorlesung und selbständiges Bearbeiten von Übungsaufgaben; ergänzende Literatur zum Selbststudium				
<b>Bezug zu anderen Fächern/Modulen:</b> Die Inhalte stehen im Zusammenhang mit den Mathematik- Modulen sowie allen anderen Modulen, die numerische Algorithmen benötigen wie z.B. das Modul "Graphische DV und Bildverarbeitung".				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Thomas Huckle, Stefan Schneider: Numerische Methoden. Springer Verlag</li> <li>• Michael Knorrenschild: Numerische Mathematik. Fachbuchverlag Leipzig</li> <li>• Vorlesungsfolien</li> </ul>				
<b>Dozenten:</b> verschiedene				
<b>Modulverantwortliche:</b> Ueberholz				
<b>Aktualisiert:</b> 23.05.2018				

<b>Modul</b>	<b>WPS Seminarmodul</b>			<b>Credits: 4</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Wahlpflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	4. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	90	
<b>Übung</b>	0	0	0	
<b>Praktikum</b>	0	0	0	
<b>Arbeitsaufwand in Stunden</b>		30	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> abhängig vom fachlichen Themengebiet				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> Testat				
<b>Notensystem:</b> bestanden / nicht bestanden				
<b>Lernziele/Kompetenzen:</b> Die / der Studierende kann <ul style="list-style-type: none"> <li>• einen fachwissenschaftlichen Vortrag erarbeiten und präsentieren,</li> <li>• vor Fachpublikum ein wissenschaftliches Thema diskutieren,</li> <li>• eine Ausarbeitung des Seminarvortrags erstellen,</li> <li>• Fachliteratur recherchieren und verwenden und</li> <li>• Präsentationssoftware sowie -techniken handhaben</li> </ul>				
<b>Inhalte:</b> Im Seminar werden Thematiken der Module des Studiengangs bzw. Thematiken, die in enger Verbindung mit den Modulinhalten stehen, behandelt. Spezielle Inhalte des Studiengangs werden vertieft bzw. erweitert. Jeder teilnehmende Studierende erarbeitet nach Vorgabe des Themas durch den Lehrenden einen Seminarvortrag, trägt ihn den anderen Seminarteilnehmern vor und fertigt eine schriftliche Ausarbeitung an. Die vorgetragenen Inhalte stehen im Anschluss des Vortrags zur Diskussion. Im Rahmen der Seminars werden Vorträge der wissenschaftlichen Vortragsreihe des Fachbereichs besucht.				
<b>Lehrmethoden:</b> Vorträge der teilnehmenden Studierenden und externer Referenten, schriftliche Ausarbeitung des Seminarvortrags und Diskussion				
<b>Bezug zu anderen Fächern/Modulen:</b> abhängig vom fachlichen Themenbereich				
<b>Literatur:</b> abhängig vom fachlichen Themenbereich				
<b>Dozenten:</b> mehrere				
<b>Modulverantwortliche:</b> Meuser				
<b>Aktualisiert:</b> 20.05.2018				

<b>Modul</b>	<b>PRM Projektmanagement</b>			<b>Credits: 3</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	4. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	30	
<b>Übung</b>	1	15	15	
<b>Praktikum</b>	0	0		
<b>Arbeitsaufwand in Stunden</b>		45	45	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> keine				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> Testat				
<b>Notensystem:</b> bestanden / nicht bestanden				
<p><b>Lernziele/Kompetenzen:</b> Vermittlung der Grundlagen des Projektmanagements zur Bildung anwendungsorientierter Fach- und Methodenkompetenz und zur Verbesserung der sozialen Kompetenz hinsichtlich der Befähigung zur Mitwirkung in oder Leitung von internationalen und gemischtgeschlechtlichen Gruppen. Nach der Teilnahme an diesem Modul ist der Studierende in der Lage,</p> <ul style="list-style-type: none"> <li>• Projekte systematisch zu strukturieren und zeitlich zu planen,</li> <li>• Ressourcen einzusetzen und</li> <li>• den Einsatz von Mitarbeitern sozial verantwortlich zu planen und zu begleiten.</li> </ul>				
<p><b>Inhalte:</b> Grundlagen</p> <ul style="list-style-type: none"> <li>• Projektorganisation</li> <li>• Projektdefinition und -strukturplanung</li> <li>• Informations-, Dokumentations- und Konfigurationsmanagement</li> <li>• Zeitplanung</li> <li>• Kostenplanung und Budgetierung</li> <li>• Ressourcenplanung, insbesondere Human Resource Management</li> <li>• Projektkontrolle und -beendigung</li> <li>• Risikomanagement</li> <li>• EDV-gestütztes Projektmanagement</li> </ul>				
<p><b>Lehrmethoden:</b> Vorlesung mit Skript und Literatur zum Selbststudium; Bearbeitung von Aufgabenstellungen in den Übungsstunden und in den begleitenden Online-Übungen Bezug zu anderen Fächern/Modulen: Das Modul gehört zu den Modulen des Themengebiets Software-Engineering. Die Lernergebnisse können in den weiteren Modulen, insbesondere dem Modul Software-Engineering sowie den Projektmodulen unmittelbar genutzt werden</p>				
<p><b>Bezug zu anderen Fächern/Modulen:</b> Das Modul gehört zu den Modulen des Themengebiets Software-Engineering. Die Lernergebnisse können in den weiteren Modulen, insbesondere dem Modul Software-Engineering sowie den Projektmodulen unmittelbar genutzt werden</p>				
<b>Literatur:</b> Online-Literatur: PMI Handbook: A Guide to the Project Management Body of Knowledge				
<b>Dozenten:</b> Göpel-Gruner				
<b>Modulverantwortliche:</b> Göpel-Gruner				
<b>Aktualisiert:</b> 22.05.2018				

<b>Modul</b>	<b>JUR Recht in der Berufspraxis des Ingenieurs</b>		<b>Credits: 2</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	4. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	30
<b>Übung</b>			
<b>Praktikum</b>			
<b>Arbeitsaufwand in Stunden</b>		30	30
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> keine			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> Testat			
<b>Notensystem:</b>			
<b>Lernziele/Kompetenzen:</b> Die / der Studierende kennt die rechtlichen Rahmenbedingungen des Berufslebens als angestellter oder selbstständiger Ingenieur.			
<b>Inhalte:</b> Einführung: Recht als Fundament beruflicher Betätigung; <ul style="list-style-type: none"> <li>• Haftung und Verantwortung: Pflichtverletzung, Verschulden und Haftung;</li> <li>• Vertragstypen: Rechte und Pflichten bei Kauf, Miete, Werk- und Dienstvertrag;</li> <li>• Arbeitsrecht: Kündigung und Befristung des Arbeitsvertrags, Arbeitszeugnisse;</li> <li>• Softwareurheberrecht: Verwertungsrechte, Einräumung von Nutzungsrechten;</li> <li>• Patentrecht: Patentfähigkeit, Wirkungen des Patents, Rechte aus dem Patent;</li> <li>• Markenrecht: Schutzvoraussetzungen, Wirkung des Markenschutzes;</li> <li>• Recht des elektronischen Geschäftsverkehrs: Verantwortung und Pflichten im Internet;</li> <li>• Handelsrecht: Kaufmannseigenschaft, Handelskauf, Handelsvertretung;</li> <li>• Steuerrecht: Einblick in die Steuerpflichten eines Selbstständige;</li> </ul>			
<b>Lehrmethoden:</b> Vorlesung, Diskussion			
<b>Bezug zu anderen Fächern/Modulen:</b> ergänzt die übrigen Fächer um außerfachliche Kenntnisse für das Berufsleben			
<b>Literatur:</b> einschlägige Gesetze			
<b>Dozenten:</b> Keller			
<b>Modulverantwortliche:</b> Keller			
<b>Aktualisiert:</b> 12.06.2018			

<b>Modul</b>	<b>WPV2 Vorlesungsmodul 2</b>		<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Wahlpflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	5. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	60
<b>Übung</b>	2	30	30
<b>Praktikum</b>	0	0	0
<b>Arbeitsaufwand in Stunden</b>		60	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Abhängig von den fachlichen Inhalten; bei Ergänzungsfächern werden nur allgemeine Informatik-Kenntnisse vorausgesetzt, bei Vertiefungsfächern werden die Inhalte der entsprechenden Hauptfachlehrveranstaltungen vorausgesetzt.			
<b>Prüfungsvorleistung:</b>			
<b>Prüfungsform:</b> nach Modulbeschreibung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Es sollen das Pflichtprogramm ergänzende oder vertiefende anwendungsorientierte Kenntnisse und Fähigkeiten in der Informatik vermittelt werden. Dabei liegt das Schwergewicht auf der Vermittlung von Methodenkompetenzen zur Entwicklung von Problemlösungskonzepten.			
<b>Inhalte:</b> abhängig vom fachlichen Themenbereich (s. Bsp.-Module der folgenden Seiten).			
<b>Lehrmethoden:</b> Vorlesung mit Herleitung der wesentlichen Kenntnisse; Vor- und Nachbereitung anhand der Referenzen. Übungen; in der Übung angeleitetes Lösen von Aufgaben, zusätzliche Hausübungen			
<b>Bezug zu anderen Fächern/Modulen:</b> Die Inhalte der Vertiefungsfächer sind sorgfältig mit denen der Hauptfachlehrveranstaltungen abgestimmt und ergänzen bzw. vertiefen die Kenntnisse und Fähigkeiten der Studierenden.			
<b>Literatur:</b> abhängig vom fachlichen Themenbereich			
<b>Dozenten:</b> verschiedene			
<b>Modulverantwortliche:</b> Meuser			
<b>Aktualisiert:</b> 20.05.2018			

<b>Modul</b>	<b>SWE Softwareengineering</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	5. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	30	
<b>Übung</b>	2	30	30	
<b>Praktikum</b>	1	15	15	
<b>Arbeitsaufwand in Stunden</b>		75	75	

**Zulassungsvoraussetzungen:** wie in der Prüfungsordnung angegeben

**Vorkenntnisse:** Der Studierende beherrscht die objektorientierte Anwendungsentwicklung (Programmiersprache C++), kann webbasierte Anwendungssysteme konzipieren und implementieren (HTML, CSS, Programmiersprachen Javascript und Python), analysiert, entwirft und implementiert interaktive Systeme und beherrscht die Grundlagen des Projektmanagements.

**Prüfungsvorleistung:** wie in der Prüfungsordnung angegeben

**Prüfungsform:** schriftliche benotete Prüfung

**Notensystem:** deutsche Notenskala 1-5

**Lernziele/Kompetenzen:** Absolventen des Moduls

- sind vertraut mit Lösungsmustern
- erkennen und formulieren funktionale und nichtfunktionale Anforderungen
- definieren Schnittstellen so, dass die Systeme wartbar, erweiterbar und zuverlässig sind
- können abstrahieren
- kennen den logischen Aufbau von IT-Systemen und deren Funktionen
- kennen Software-Architekturen
- können Muster und Bibliotheken einsetzen
- erstellen und testen größere IT-Systeme
- kennen Entwicklungsprozesse
- besitzen Grundkenntnisse im Schätzen und Messen von Aufwand und Produktivität
- sind kontaktfähig und arbeiten in Gruppen
- können einem Text wichtige Inhalte entnehmen, diese strukturieren und wiedergeben
- nutzen verschiedene Medien zur Informationsbeschaffung
- können vorhandene Missverständnisse zwischen Gesprächspartnern erkennen und abbauen
- führen ein Ziel und Zeitmanagement aus

**Inhalte:** Requirements Management zum systematischen Erkennen und Analysieren sowie zur Verfolgung von Anforderungen bei der Software-Entwicklung, Qualitätsmanagement : Grundlagen, Normen, QM-systeme, Software-Prozessmanagement : Grundlagen von Prozessmodellen, vergleichende Bewertung, kontinuierliche Prozessverbesserung, CMMI, Einbettung der Ergebnisse der weiteren Veranstaltungen (s.u.) in den Gesamtkontext des Software-Engineering zur Durchführung von Software-Entwicklungsprojekte

**Lehrmethoden:** Vorlesung mit Skript und Literatur zum Selbststudium; Bearbeitung von Aufgabenstellungen in den Übungsstunden; Bearbeitung eines Software-Entwicklungsprojekts in kleinen Teams mit theoretischer Vorbereitung, praktischer Einführung in die Arbeitsumgebung, Anforderungsanalyse und Entwurf, einschließlich des bedarfsgerechten Entwurfs der Benutzungsschnittstelle sowie eines Projekts zur Realisierung eines Werkzeugs zur Unterstützung des Software-Entwicklungs-Prozesses.

**Bezug zu anderen Fächern/Modulen:** Das Modul schließt das Themengebiet Software-Engineering ab, zu dem auch die Module Objektorientierte Anwendungsentwicklung, Interaktive Systeme und Projektmanagement zählen. Das Modul behandelt insbesondere die Themen des Software-Engineering, die durch die genannten Module nicht abgedeckt werden.

<b>Literatur:</b> <ul style="list-style-type: none"><li>• Balzert: Software-Technik Band 1 und 2, W3L</li><li>• Sommerville: Software-Engineering, Pearson Studium</li><li>• Ebert: Requirements Management, dpunkt-Verlag</li><li>• Siedersleben: Software-Architektur, dpunkt-Verlag</li><li>• Ludwig / Lichter: Software-Engineering, dpunkt-Verlag</li></ul>
<b>Dozenten:</b> Beims
<b>Modulverantwortliche:</b> Beims
<b>Aktualisiert:</b> 25.05.2018

<b>Modul</b>	<b>ITS IT-Sicherheit</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>				
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	5. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	60	
<b>Übung</b>	1	15	15	
<b>Praktikum</b>	1	15	15	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Betriebssysteme, IP-Kommunikation				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Nach Teilnahme an diesem Modul ist der Studierende in der Lage, <ul style="list-style-type: none"> <li>• die Gefährdung in einem IT-System (Rechner, Netzwerk) zu analysieren (Risikoanalyse),</li> <li>• Maßnahmen zur Abwehr zu konzipieren,</li> <li>• sichere Netzstrukturen aus Hard- und Software im Hinblick auf IT-Sicherheit zu entwerfen,</li> <li>• IT-Systeme geeignet abzusichern,</li> <li>• Software unter Berücksichtigung von IT-Sicherheit zu entwerfen und zu realisieren und</li> <li>• geeignete Maßnahmen im Fall eines Angriffes zu ergreifen.</li> </ul>				
<b>Inhalte:</b> Praxisorientierte Einführung in die Rechner- und Netzwerksicherheit. Erläuterung des rechtlichen Rahmens, Schutzziele (Integrität, Vertraulichkeit, Verfügbarkeit), Gefährdungspotenzial, Risikoanalyse. Einführung in die Kryptografie (symmetrische, asymmetrische Verschlüsselung, PKI). Angriffstechniken (Brute-Force-Attacken, Buffer-Overflow, Viren, Würmer, Trojaner, Phishing). Abwehrmaßnahmen: strukturelle Maßnahmen über dedizierte Sicherheitsarchitekturen (zum Beispiel demilitarisierte Zonen, Virtual Private Networks), Einsatz aktiver Komponenten, Firewall, Virenabwehr, IT-Sicherheit für Programmierer. Sicherheit von Betriebssystemen.				
<b>Lehrmethoden:</b> Rechnergestützte Vorlesung mit Skript zum Selbststudium; Praktikumsvorbereitung über "Hackits"; Übung am eigenen oder zur Verfügung gestellten Notebook (verschlüsselte Datenablage, verschlüsselte EMail-Kommunikation); Laborversuche zur Sicherheit (sicheres WLAN, Capture the Flag, Angriff und Sicherung von Industrieanlagen/IoT)				
<b>Bezug zu anderen Fächern/Modulen:</b> Die Veranstaltung ergänzt die Vorlesungen Betriebssysteme und Datennetze.				
<b>Literatur:</b> J. Quade: Rechner- und Netzwerksicherheit <ul style="list-style-type: none"> <li>• Skript zur Vorlesung, jeweils aktuelle Auflage</li> </ul>				
<b>Dozenten:</b> Quade, Meuser				
<b>Modulverantwortliche:</b> Quade				
<b>Aktualisiert:</b> 27.02.2018				

<b>Modul</b>	<b>EZS Echtzeitsysteme</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>				
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	5. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	60	
<b>Übung</b>	1	15	15	
<b>Praktikum</b>	1	15	15	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Rechnerarchitekturen, Programmierkenntnisse in C, Betriebssystemarchitekturen, Task-Management (Scheduling), Memory- und I/O-Management, Basiskenntnisse in Systemadministration				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> schriftliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Nach Teilnahme an diesem Modul ist der Studierende in der Lage, <ul style="list-style-type: none"> <li>• zeitliche Anforderungen technischer Systeme zu erkennen und zu charakterisieren,</li> <li>• unterschiedliche Lösungsansätze für Aufgaben mit zeitlichen Anforderungen (auch im sicherheitstechnischen Umfeld) zu konzipieren, zu bewerten und zu realisieren,</li> <li>• technische Prozesse softwaretechnisch an Rechnersysteme anzukoppeln und</li> <li>• die Realzeitfähigkeit mit Hilfe mathematischer Verfahren nachzuweisen.</li> <li>• Der Studierende kennt Probleme und Lösungen (u.a. moderne, standardisierte Programmierschnittstellen) beim programmtechnischen Umgang von Zeiten,</li> <li>• Architektur, Konzepte und Funktionsweisen moderner Realzeitsysteme.</li> <li>• Er versteht das Zusammenspiel von Hard- und Software</li> </ul>				
<b>Inhalte:</b> Echtzeitbetrieb und schritthaltende Verarbeitung; zentrale Beschreibungsgrößen von Realzeitsystemen, Realzeitbedingungen, Systemaspekte. Systemsoftware, insbesondere Realzeitbetriebssysteme inklusive Scheduling und IO-Management. Aspekte der nebenläufigen Realzeitprogrammierung: Taskmanagement, kritische Abschnitte, Umgang mit Zeiten, Inter-Prozess-Kommunikation, Peripheriezugriffe, Bitoperationen. Realzeitarchitekturen unter anderem mit Threaded Interrupts, RT-Multicore- und Multikernel-Architekturen. Betriebssicherheit (Safety) und Verfügbarkeit. Formale Beschreibungsmethoden für Realzeitsysteme. Realzeitnachweis bei Einsatz von prioritätengesteuertem- und Deadline-Scheduling.				
<b>Lehrmethoden:</b> Rechnergestützte Vorlesung; Rechnen von Aufgaben in den Übungsstunden; Online-Praktikumsvorbereitung mit automatischer Selbstkontrolle; Durchführung von im Selbststudium vorbereiteten Aufgaben im Labor; Anfertigung von Laborausarbeitungen				
<b>Bezug zu anderen Fächern/Modulen:</b> Die Vorlesung baut auf den Inhalten der Module "Technische Grundlagen der Informatik" und "Betriebssysteme" auf.				
<b>Literatur:</b> J. Quade, M. Mächtel: Moderne Realzeitsysteme kompakt. Eine Einführung mit Embedded Linux. dpunkt.Verlag, Heidelberg 2012				
<b>Dozenten:</b> Quade				
<b>Modulverantwortliche:</b> Quade				
<b>Aktualisiert:</b> 26.02.2018				

<b>Modul</b>	<b>ITS Projektmodul IT-Mobile IT mit Android</b>		<b>Credits: 6</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	5. Semester		inkl. Prüfungsvorbereitung
<b>Sem. Lehrveranstaltung</b>		0	0
<b>Praktikum</b>	4	60	120
<b>Arbeitsaufwand in Stunden</b>		60	120
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Die Studierenden verfügen über Grundkenntnisse in der objektorientierten Anwendungsentwicklung und im Projekt-Management.			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> Testat; Projektarbeit, schriftlicher Projektbericht und Präsentation			
<b>Notensystem:</b> bestanden / nicht bestanden			
<b>Lernziele/Kompetenzen:</b> Der Studierende <ul style="list-style-type: none"> <li>• kann Anwendungsfelder für mobile IT-Lösungen charakterisieren;</li> <li>• kennt die besonderen Herausforderungen (Beschränkungen bei CPU-Leistung, Speicher, Bildschirmgröße, Konnektivität) und Potentiale (Orts- und Kontextsensitivität) mobiler Apps;</li> <li>• ist in der Lage, mithilfe des Android-Frameworks eine mobile App zu konzipieren und umzusetzen</li> <li>• und dabei koordiniert und arbeitsteilig mit anderen Projektpartnern zusammenarbeiten,</li> <li>• Methoden der Projekt-Managements (Meilenstein-Planung/Controlling) anzuwenden,</li> <li>• eine technische Dokumentation anfertigen und</li> <li>• die Ergebnisse vor einem Fachpublikum präsentieren.</li> </ul>			
<b>Inhalte:</b> Fachlich: Mobile IT-Lösungen; App-Programmierung mit dem Android Framework <ul style="list-style-type: none"> <li>• Methodisch: Projekt-Planung und -Controlling; Arbeitsteilung; Kommunikation und Konflikt-Management; Präsentation von Ergebnissen</li> </ul>			
<b>Lehrmethoden:</b> Kurtutorials zum Einstieg in das Android-Framework, Gruppenarbeit, selbständiges Arbeiten			
<b>Bezug zu anderen Fächern/Modulen:</b> Anhand eines größeren Projekts vertiefen die Studierenden die in den Fächern Objektorientierte Anwendungsentwicklung und Projektmanagement erworbenen Fähigkeiten. Je nach gewähltem Projekt werden auch Kompetenzen aus anderen Modulen (z.B. IT-Sicherheit oder Graphische Datenverarbeitung) benötigt und vertieft			
<b>Literatur:</b> Thomas Künath: Android 8: Android 8: Das Praxisbuch für Java-Entwickler. Rheinwerk Computing, 2018. <ul style="list-style-type: none"> <li>• Bill Phillips: Android Programming: The Big Nerd Ranch Guide (3rd ed.), 2017.</li> </ul>			
<b>Dozenten:</b> Weidenhaupt			
<b>Modulverantwortliche:</b> Weidenhaupt			
<b>Aktualisiert:</b> 05.06.2018			

<b>Modul</b>	<b>RGA Rechtliche und gesellschaftliche Aspekte der Informatik Credits: 3</b>		
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	5. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	3	45	45
<b>Übung</b>			
<b>Praktikum</b>			
<b>Arbeitsaufwand in Stunden</b>		45	45
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> keine			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> Testat			
<b>Notensystem:</b> bestanden / nicht bestanden			
<b>Lernziele/Kompetenzen:</b> Absolventen dieses Moduls <ul style="list-style-type: none"> <li>• können ethische Bewertungen auf Basis ethischer Theorien entwickeln und begründen</li> <li>• können die Gesetze zum Persönlichkeitsrecht benennen und erkennen elementare Verstöße</li> <li>• Je nach Themenwahl der Studierenden können Sie ferner,</li> <li>• die Hintergründe des Begriffs "Intellectual Property" erläutern, die diesbezüglichen Gesetze und deren praktischen Auswirkungen benennen, und die durch verbreitete Lizenzmodelle gewährten Rechte angeben</li> <li>• Beispiele für unethisches Verhalten in der Wirtschaft nennen und Maßnahmen zu dessen Vermeidung vorschlagen</li> </ul>			
<b>Inhalte:</b> Die rasante Entwicklung der Informationstechnik hat mittlenweile viele Anwendungsgebiete erschlossen, die gesellschaftliche Probleme aufwerfen. Um solche Anwendungen zu bewerten sind Kriterien dafür nötig, was "gut" ist, was die Aufgabenstellung der "Ethik" ist. <ul style="list-style-type: none"> <li>• Folglich beginnt die Veranstaltung mit einer Einführung in verschiedene theoretische Ethikansätze. Im Licht dieser Ansätze werden dann aktuelle Themen diskutiert, wie z.B. Fragen des Datenschutzes, der "Intellectual Property"</li> <li>• (Digital Rights Management, Copyleft-Lizenzen, Softwarepatente etc.) oder der Wirtschaftskriminalität. Dazu werden auch die entsprechenden Gesetzestexte gelesen und anhand von Beispielen diskutiert.</li> </ul>			
<b>Lehrmethoden:</b> Dozentenvortrag, Diskussion der Themen anhand von Fallstudien, Selbstarbeitsphasen in Gruppenarbeit			
<b>Bezug zu anderen Fächern/Modulen:</b> Die Veranstaltung ergänzt die übrigen Module um einen dort nicht thematisierten Aspekt: die ethische Bewertung informationstechnischer Anwendungen. Somit ist dieses Modul eine wichtige Ergänzung für die Persönlichkeitsbildung der Studierenden.			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Michael Quinn: Ethics for the Information Age. Addison-Wesley 2005</li> <li>• Der Bundesbeauftragte für den Datenschutz: Bundesdatenschutzgesetz - Text und Erläuterung. (September 2007)</li> <li>• Jeanette Hofmann (Hrsg.): Wissen und Eigentum. Bundeszentrale für politische Bildung (2006)</li> <li>• J. Boyle: The Public Domain - Enclosing the Commons of the Mind. Yale University Press (2008)</li> <li>• Die aktuellen Gesetzestexte können der Webseite des Bundesjustizministeriums entnommen werden</li> </ul>			
<b>Dozenten:</b> Göpel-Gruner, Dalitz			
<b>Modulverantwortliche:</b> Dalitz			
<b>Aktualisiert:</b> 11.05.2018			

<b>Modul</b>	<b>WPV2 Netzwerksicherheit</b>			<b>Credits: 5</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Wahlpflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	5. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	2	30	60	
<b>Übung</b>	2	30	30	
<b>Praktikum</b>	0	0	0	
<b>Arbeitsaufwand in Stunden</b>		60	90	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> Grundlagen Datennetze, TCP/IP-Protokolle, IP-Adressierung und Adressverwaltung; Routing und Router-Programmierung, Switching in LANs, Switchprogrammierung, WAN-Technologien, Access-Listen				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> benotete Prüfung - Abschlussarbeit				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Vermittlung anwendungsorientierter Kenntnisse und Fähigkeiten in der Netzwerksicherheit, der Kompetenzen zur Analyse von Sicherheitsproblemen, der Kompetenzen für den Entwurf und der Realisierung einer Sicherheitsarchitektur, technologischer Kompetenzen für die Entwicklung und den Einsatz von Schutzmechanismen und einer umfassenden Methodenkompetenz zur Entwicklung von Problemlösungskonzepten. Absolventen dieses Moduls sind in der Lage, <ul style="list-style-type: none"> <li>• Konzepte und Funktionsweisen von Architekturen zur Netzwerksicherheit zu verstehen</li> <li>• Unternehmensnetze mit der entsprechenden Hard- und Software zu implementieren, die die Anforderungen an einen sicheren Betrieb vollständig erfüllen</li> <li>• ein Bewusstsein für die vielfältige Sicherheitsprobleme zu entwickeln, sowie Techniken und Verfahren zur Sicherung von Unternehmensnetzen einzusetzen</li> <li>• Sicherheitsprobleme im Netzwerk zu erkennen, Gegenmaßnahmen zu konzipieren und eine Problemlösung zu implementieren.</li> </ul>				
<b>Inhalte:</b> Netzwerksicherheit allgemein (Angriffsszenarios, Schutzkriterien, Schutzkonzepte); Absicherung der Netzwerkgeräte; Authentifizierung, Autorisierung und Abrechnung, Firewall-Technologien; IPS/IDS-Implementierung; Kryptographie; VPNs; Security-Appliances, Implementierung einer Unternehmenssicherheit				
<b>Lehrmethoden:</b> Vorlesung, Übungen im Laborraum, zusätzliche praktische Übungen im Lernmodul Packet Tracer, Online-Tests, Vor- und Nachbereitung aller Veranstaltungen und Klausurvorbereitung mit Online-Curriculum				
<b>Bezug zu anderen Fächern/Modulen:</b> Zusammen mit den Inhalten der Moduls Datennetze und Datenübertragung und des Moduls Datennetzmanagement sind die Kenntnisse und Fähigkeiten zu erlangen, die zum Erwerb des Industriezertifikats CCNA Security für Spezialisten in Netzwerksicherheit erforderlich sind				
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Cisco Press: CCNA Security Course Booklet Version 2, 2015 (ISBN: 978-1587133510)</li> <li>• CCNA Security Lab Manual Version 2, 2015 (ISBN: 978-1587133503)</li> <li>• Alexander Michael: Netzwerke und Netzwerksicherheit, Hüthing Verlag, 2006 (ISBN: 978-3826650482)</li> </ul>				
<b>Dozenten:</b> Meuser				
<b>Modulverantwortliche:</b> Meuser				
<b>Aktualisiert:</b> 20.05.2018				

<b>Modul</b>	<b>WPV2 Logikprogrammierung und funktionale Programmierung Credits: 5</b>		
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Wahlpflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Sommersemester		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	5. Semester		inkl. Prüfungsvorbereitung
<b>Vorlesung</b>	2	30	60
<b>Übung</b>	2	30	30
<b>Praktikum</b>	0	0	0
	<b>Arbeitsaufwand in Stunden</b>	60	90
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Logik und Umgang mit Funktionen im Umfang der Vorlesungen Mathematik 1 und 2, elementare Programmierkenntnisse			
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben			
<b>Prüfungsform:</b> mündliche benotete Prüfung			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Kennenlernen eines ergänzenden Programmierparadigmas <ul style="list-style-type: none"> <li>• Die Studierenden sind in der Lage, das objektorientierte bzw. prozedurale Programmiermodell in einem allgemeineren Kontext zu sehen.</li> <li>• Die Studierenden sind in der Lage, abstrakte mathematische Inhalte für Anwendungen zu nutzen.</li> <li>• Sie können Aufgabenstellungen logisch und strukturiert analysieren.</li> <li>• Sie können die erlernten Programmier Techniken auch in anderen Programmiersprachen anwenden</li> </ul>			
<b>Inhalte:</b> Beim Programmieren in C++ oder Java muss man für ein Problem Lösungsalgorithmen entwickeln. Dabei ist explizit zu beschreiben, WIE das Problem zu lösen ist. Bei einer deklarativen Programmierung beschreibt man dagegen exakt das Problem (das WAS) und gibt keinen Lösungsalgorithmus vor. SQL ist beispielsweise deklarativ. Ebenfalls deklarativ ist die Programmiersprache PROLOG, die universell einsetzbar ist. Ein Problem wird hier mittels Fakten und Regeln (und damit mittels mathematischer Logik) beschrieben. Diese Herangehensweise eignet sich besonders gut für Expertensysteme. Inhalte des Moduls sind: <ul style="list-style-type: none"> <li>• Aussagenlogik, Resolutionskalkül</li> <li>• Prädikatenlogik</li> <li>• Einführung in PROLOG</li> <li>• Techniken der Logikprogrammierung</li> <li>• Ideen der funktionalen Programmierung</li> <li>• Kurzer Einblick in funktionale Programmiersprachen wie z. B. Lisp und Erlang</li> </ul>			
<b>Lehrmethoden:</b> Vorlesung mit Herleitung der wesentlichen Kenntnisse; Vor- und Nachbereitung anhand der Referenzen. Übungen; in der Übung angeleitetes Lösen von Aufgaben am Computer			
<b>Bezug zu anderen Fächern/Modulen:</b> Die Inhalte von Mathematik 1 und 2 werden hinsichtlich der Aussagen- und Prädikatenlogik aufgegriffen und erweitert. Programmierkenntnisse im Rahmen der "Einführung in die Programmierung" werden benötigt			
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• P. Forbig, I. O. Kerner: Programmierung - Paradigmen und Konzepte. Fachbuchverlag Leipzig/Carl Hanser, München, 2006.</li> <li>• W.F. Clocksin, C.S. Mellish: Programming in Prolog. Springer, New York, 1981.</li> <li>• L. Sterling, E. Shapiro: Prolog - Fortgeschrittene Programmier Techniken. Addison-Wesley, Bonn, 1988.</li> <li>• I. Bratko: Prolog - Programming for Artificial Intelligence. Addison Wesley, 2000</li> <li>• L. Piepmeyer: Grundkurs Funktionale Programmierung mit Scala, Hanser, München, 2010.</li> <li>• J. Armstrong: Programming in Erlang, Pragmatic Bookshelf, Raleigh, North Carolina, 2007.</li> </ul>			
<b>Dozenten:</b> Goebbels			
<b>Modulverantwortliche:</b> Goebbels			
<b>Aktualisiert:</b> 08.04.2018			

<b>Modul</b>	<b>PRX Praxisphase</b>			<b>Credits:</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>				
<b>Turnus des Angebots</b>				
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	6. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	1	8	16	
<b>Übung</b>				
<b>Praktikum</b>		420		
<b>Arbeitsaufwand in Stunden</b>		428	16	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> abhängig vom Projekt				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> Testat				
<b>Notensystem:</b> bestanden / nicht bestanden				
<b>Lernziele/Kompetenzen:</b> In diesem Modul zeigt der/die Studierende, dass er/sie <ul style="list-style-type: none"> <li>• sich in bestehende Arbeitszusammenhänge einfügen kann,</li> <li>• kooperativ in Teams arbeitet und darin zielorientiert argumentieren und mit Kritik umgehen kann,</li> <li>• verschiedene Medien zur Informationsbeschaffung nutzen kann,</li> <li>• Projektaufgaben aus dem beruflichen Alltag eines Informatikers lösen kann,</li> <li>• sein theoretisches Wissen zur Analyse und Lösung von praktischen Aufgabenstellungen einsetzen kann,</li> <li>• Ideen und Lösungsvorschläge präsentieren und diskutieren kann und</li> <li>• die eigene Arbeit in Form eines schriftlichen Berichts dokumentieren kann.</li> </ul>				
<b>Inhalte:</b> Durchführung von Projekten oder Teilprojekten aus der Praxis von Informatikern oder Elektrotechnik-Ingenieuren				
<b>Lehrmethoden:</b> selbständiges Arbeiten, Projektarbeit, Gruppenarbeit				
<b>Bezug zu anderen Fächern/Modulen:</b> abhängig vom Projekt				
<b>Literatur:</b> abhängig vom Projekt				
<b>Dozenten:</b> alle Lehrenden				
<b>Modulverantwortliche:</b> Dalitz				
<b>Aktualisiert:</b> 11.05.2018				

<b>Modul</b>	<b>BA Bachelorarbeit</b>		<b>Credits: 12</b>
<b>Studiengang</b>	Bachelor Informatik		
<b>Modultyp</b>	Pflichtmodul		
<b>Sprache</b>	Deutsch		
<b>Turnus des Angebots</b>	Jedes Studienjahr		
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>
	6. Semester		inkl. Prüfungsvorbereitung
<b>Sem. Lehrveranstaltung</b>			
<b>Praktikum</b>		180	180
<b>Arbeitsaufwand in Stunden</b>		180	180
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben			
<b>Vorkenntnisse:</b> Fähigkeit zur selbständigen ingenieurmäßigen Arbeit			
<b>Prüfungsvorleistung:</b>			
<b>Prüfungsform:</b> benotete Prüfung - Abschlussarbeit			
<b>Notensystem:</b> deutsche Notenskala 1-5			
<b>Lernziele/Kompetenzen:</b> Die / der Studierende kann <ul style="list-style-type: none"> <li>• eine Aufgabenstellung aus der Informatik unter Anwendung des im Studium erlernten Fachwissens sowie wissenschaftlicher Methoden und Erkenntnisse innerhalb einer vorgegebenen Frist selbständig bearbeiten,</li> <li>• die Ergebnisse in fachliche und fächerübergreifende Zusammenhänge einordnen, in Form einer ingenieurwissenschaftlichen Arbeit darstellen und vor sachkundigem Publikum präsentieren.</li> </ul>			
<b>Inhalte:</b> Analyse der Problemstellung und Abgrenzung des Themas, Literatur-/Patentrecherche, Formulierung des Untersuchungsansatzes/der Vorgehensweise, Festlegung eines Lösungskonzepts bzw. -wegs, Planung und Erarbeitung der Lösung, Analyse der Ergebnisse, Einschätzung der Bedeutung für die Praxis, Zeitmanagement; Darstellung der Arbeitsergebnisse in Form einer ingenieurwissenschaftlichen Arbeit; Präsentation der Ergebnisse vor sachkundigem Publikum; es wird verlangt, dass bei der Durchführung der Arbeit die wissenschaftliche Arbeitsweise und Methodik Anwendung findet; systematisch, analytisch und methodisch korrekt vorgegangen, logisch und prägnant argumentiert sowie zielorientiert und zeitkritisch gearbeitet wird und die Arbeitsergebnisse formal korrekt dargestellt und überzeugend verteidigt werden können. Für die Abschlussarbeit ist eine Bearbeitungszeit von 12 Wochen und für das anschließende Kolloquium eine Vorbereitungszeit von 3 Wochen vorgesehen.			
<b>Lehrmethoden:</b> Anleitung zum wissenschaftlichen Arbeiten			
<b>Bezug zu anderen Fächern/Modulen:</b> abhängig von der Thematik der Bachelorarbeit; anschließendes Kolloquium zur Präsentation und Verteidigung der Ergebnisse der Abschlussarbeit			
<b>Literatur:</b> abhängig von der Thematik der Bachelorarbeit			
<b>Dozenten:</b> alle Lehrenden			
<b>Modulverantwortliche:</b> alle Lehrenden			
<b>Aktualisiert:</b> 05.06.2018			

<b>Modul</b>	<b>KOL Kolloquium</b>			<b>Credits: 3</b>
<b>Studiengang</b>	Bachelor Informatik			
<b>Modultyp</b>	Pflichtmodul			
<b>Sprache</b>	Deutsch			
<b>Turnus des Angebots</b>	Jedes Studienjahr			
	<b>Semesterwochenstunden</b>	<b>Präsenzzeit</b>	<b>Selbststudium</b>	
	6. Semester		inkl. Prüfungsvorbereitung	
<b>Vorlesung</b>	1	1	89	
<b>Übung</b>				
<b>Praktikum</b>				
<b>Arbeitsaufwand in Stunden</b>		1	89	
<b>Zulassungsvoraussetzungen:</b> wie in der Prüfungsordnung angegeben				
<b>Vorkenntnisse:</b> abhängig von der Bachelorarbeit				
<b>Prüfungsvorleistung:</b> wie in der Prüfungsordnung angegeben				
<b>Prüfungsform:</b> mündliche benotete Prüfung				
<b>Notensystem:</b> deutsche Notenskala 1-5				
<b>Lernziele/Kompetenzen:</b> Die / der Studierende kann <ul style="list-style-type: none"> <li>• eine fachwissenschaftliche Diskussion führen und</li> <li>• Arbeitsergebnisse präsentieren und verteidigen.</li> </ul>				
<b>Inhalte:</b> Präsentation der Ergebnisse der Bachelorarbeit, Verteidigung und Diskussion der Ergebnisse im Fachgespräch				
<b>Lehrmethoden:</b> Präsentation, Gespräch				
<b>Bezug zu anderen Fächern/Modulen:</b> Bachelorarbeit				
<b>Literatur:</b>				
<b>Dozenten:</b> alle Lehrenden				
<b>Modulverantwortliche:</b> alle Lehrenden				
<b>Aktualisiert:</b> 05.06.2018				