



# **Modulhandbuch**

## **Bachelor Software Systems Science (ab WS 2015/16)**

### **Wirtschaftsinformatik und Angewandte Informatik**

**Gemäß der geltenden Fassung der Studien- und Fachprüfungsordnung vom 06.03.2015 für den Bachelorstudiengang Software Systems Science an der Otto-Friedrich-Universität Bamberg. Gültig ab Wintersemester 2020/21.**

---

## **Hinweis zur Weitergeltung älterer Fassungen eines Modulhandbuchs:**

### **1. Geltungsbeginn**

Die im vorliegenden Modulhandbuch enthaltenen Modulbeschreibungen gelten erstmals für das Semester, das auf dem Deckblatt angegeben ist.

### **2. Übergangsbestimmung**

- a. Studierende, die gemäß bisher geltendem Modulhandbuch ein Modul bereits in Teilen absolviert haben (vgl. Nr. 2b), schließen das Modul nach der bisher geltenden Fassung des Modulhandbuchs ab.

Diese Übergangsbestimmung gilt ausschließlich für den dem versäumten/nicht bestandenen/nicht absolvierten regulären Prüfungstermin unmittelbar folgenden Prüfungstermin. Auf Antrag der oder des Studierenden kann der Prüfungsausschuss in begründeten Fällen eine Verlängerung der Übergangsfrist festlegen.

- b. Ein Modul ist in Teilen absolviert, wenn die Modulprüfung nicht bestanden oder versäumt wurde. Gleiches gilt für den Fall, dass zumindest eine Modulteilprüfung bestanden, nicht bestanden oder versäumt wurde.

Ferner gilt ein Modul als in Teilen absolviert, sofern sich die oder der Studierende gemäß bisher geltendem Modulhandbuch zu einer dem jeweiligen Modul zugeordneten Lehrveranstaltung angemeldet hat.

### **3. Geltungsdauer**

Das Modulhandbuch gilt bis zur Bekanntgabe eines geänderten Modulhandbuchs auch für nachfolgende Semester.

## **Abweichungen im Modulangebot des BA Software Systems Science zur StuF-PO vom 06.03.2015, zuletzt geändert am 04.10.2017**

### **Modulgruppe A2 Allgemeine Informatik Grundlagen**

- Das Pflichtmodul MI-AuD-B hat den Modulanbieter gewechselt. Es wird fortan angeboten als AI-AuD-B.
- Das Pflichtmodul DSG-EiRBS-B hat den Modulanbieter gewechselt. Es wird fortan angeboten als PSI-EiRBS-B.
- Das Wahlpflichtmodul SEDA-DMS-B (Datenmanagementsysteme) hat den Modulanbieter gewechselt sowie in diesem Zusammenhang den Modultitel angepasst. Es wird fortan angeboten als MOBI-DBS-B (Datenbanksysteme).
- Das Wahlpflichtmodul SWT-FSA heißt fortan SWT-FSA-B.

### **Modulgruppe A3 Programmierung und Softwaretechnik**

- Das Pflichtmodul GdI-IFP heißt fortan GdI-IFP-B.
- Das Wahlpflichtmodul MOBI-IMP-B wird nicht mehr angeboten. Studierende können auf andere Module des umfangreichen Wahlkorbs in A3, Programmierung und Softwaretechnik, zurückgreifen.
- Das Modul KInf-SemInf-M „Semantic Information Processing“ wurde umbenannt in KInf-SemInf-M „Semantische Informationsverarbeitung“.

### **Modulgruppe A4 Komplexe und Verteilte Systeme**

- Das Pflichtmodul DSG-IDistrSys heißt fortan DSG-IDistrSys-B. Auch die Prüfungsform hat sich geändert: Sie lautet nun nicht mehr „Hausarbeit (3 Monate) mit Kolloquium (20 Min.)“, sondern „Hausarbeit (3 Monate) mit Kolloquium (10 Min.)“
- Das Wahlpflichtmodul KogSys-IA-B wird nicht mehr angeboten. Neu wird äquivalent das Modul AI-KI-B angeboten.
- Die Wahlpflicht-Mastermodule KTR-GIK-M und MOBI-DSC(-M) werden nicht mehr im Rahmen des Bachelor SoSySc angeboten.

## Äquivalenzliste BA Software Systems Science,

StuFPO 06.03.2015

Im Folgenden finden Sie eine Auflistung von Modulen, deren Bezeichnung bzw. Kürzel geändert wurde, ohne dass damit eine wesentliche Änderung des Moduls verbunden ist. Sofern ein in der Spalte „bisheriges Modul“ aufgeführtes Modul erfolgreich absolviert wurde, kann das in der Spalte „neues Modul“ angegebene Modul nicht belegt werden.

Die Angabe der Äquivalenzen erfolgt chronologisch.

bisheriges Modul			neues Modul		
Modulkürzel	Modulbezeichnung	bis (Semester)	Modulkürzel	Modulbezeichnung	ab (Semester)
DSG-EiRBS-B	Einführung in Rechner- und Betriebssysteme	SS18	PSI-EiRBS-B	Einführung in Rechner- und Betriebssysteme	WS18/19
MOBI-DSC	Data Streams and Complex Event Processing	SS18	MOBI-DSC-M	Data Streams and Complex Event Processing	WS18/19
SEDA-DMS-B	Datenbankmanagementsysteme	SS18	MOBI-DBS-B	Datenbanksysteme	WS18/19
KogSys-IA-B	Intelligente Agenten	SS18	AI-KI-B	Einführung in die Künstliche Intelligenz	WS18/19
SEDA-PT-B	Methoden der Präsentation, Gesprächsführung und Diskussion	WS18/19	ISM-PT-B	Methoden der Präsentation, Gesprächsführung und Diskussion	SS19
SEDA-TA-B	Technikfolgeabschätzung / -bewertung	WS18/19	ISM-TA-B	Technikfolgeabschätzung / -bewertung	SS19
DSG-IDistrSys	Introduction to Distributed Systems	SS19	DSG-IDistrSys-B	Introduction to Distributed Systems	WS19/20
Gdl-IFP	Introduction to Functional Programming	SS19	Gdl-IFP-B	Introduction to Functional Programming	WS19/20
KInf-SemInf-M	Semantic Information Processing	SS19	KInf-SemInf-M	Semantische Informationsverarbeitung	WS19/20
MI-AuD-B	Algorithmen und Datenstrukturen	WS19/20	AI-AuD-B	Algorithmen und Datenstrukturen	SS20



---

## Module

AI-AuD-B: Algorithmen und Datenstrukturen.....	9
AI-KI-B: Einführung in die Künstliche Intelligenz.....	12
DSG-AJP-B: Fortgeschrittene Java Programmierung.....	14
DSG-EiAPS-B: Einführung in Algorithmen, Programmierung und Software.....	16
DSG-IDistrSys-B: Introduction to Distributed Systems.....	19
DSG-JaP-B: Java Programmierung.....	22
DSG-PKS-B: Programmierung komplexer interagierender Systeme.....	24
DSG-Project-2-SoSySc-B: DSG Bachelorprojekt Software Systems Science.....	26
DSG-Sem-B: Bachelorseminar zur Praktischen Informatik.....	28
EESYS-IITP-B: Internationales IT-Projektmanagement.....	30
Gdl-GTI-B: Grundlagen der Theoretischen Informatik.....	32
Gdl-IFP-B: Introduction to Functional Programming.....	34
Gdl-MTL: Modal and Temporal Logic.....	36
Gdl-MfI-1: Mathematik für Informatik 1 (Aussagen- u. Prädikatenlogik).....	38
Gdl-PR2-B: Bachelorprojekt Software Systems Science.....	40
Gdl-Sem-B: Bachelorseminar Grundlagen der Informatik.....	42
HCI-IS-B: Interaktive Systeme.....	44
HCI-KS-B: Kooperative Systeme.....	46
HCI-Sem-B: Bachelorseminar Mensch-Computer-Interaktion.....	49
HCI-US-B: Ubiquitäre Systeme.....	51
ISM-PT-B: Methoden der Präsentation, Gesprächsführung und Diskussion.....	54
ISM-TA-B: Technikfolgenabschätzung/-bewertung.....	55
KInf-GeoInf-B: Geoinformationssysteme.....	57
KInf-SemInf-M: Semantische Informationsverarbeitung.....	59
KInf-Seminar-B: Bachelorseminar Kulturinformatik.....	61
KTR-Datkomm-B: Datenkommunikation.....	63
KTR-MfI-2: Mathematik für Informatik 2 (Lineare Algebra).....	67
KTR-SSSProj-B: KTR Bachelorprojekt Software Systems Science.....	69
KTR-Sem-B: Bachelorseminar zu Kommunikationssystemen und Rechnernetzen.....	72

---

# Inhaltsverzeichnis

---

KogSys-Sem-B: Bachelorseminar Kognitive Systeme.....	74
MI-Sem-B: Bachelorseminar zur Medieninformatik.....	76
MI-WAIAI-B: Einführung in das wissenschaftliche Arbeiten für Informatik und Angewandte Informatik.....	78
MI-WebT-B: Web-Technologien.....	80
MOBI-DBS-B: Datenbanksysteme.....	83
MOBI-MSS-B: Mobility in Software Systems.....	85
MOBI-PRS-B: Bachelor Project Mobile Software Systems (SoSySc).....	87
MOBI-SEM-B: Bachelor-Seminar Mobile Software Systems.....	89
PSI-EDS-B: Ethics for the Digital Society.....	90
PSI-EiRBS-B: Einführung in Rechner- und Betriebssysteme.....	92
PSI-IntroSP-B: Introduction to Security and Privacy.....	94
PSI-SSSPProject-B: Software Systems Science Project: Security and Privacy.....	96
PSI-Sem-B: Seminar Security and Privacy Foundations.....	99
SME-Phy-B: Physical Computing.....	101
SME-Sem-B: Bachelorseminar zu Smart Environments.....	103
SWT-FSA-B: Foundations of Software Analysis.....	105
SWT-FSE-B: Foundations of Software Engineering.....	107
SWT-MBT-B: Modellbasiertes Testen.....	109
SWT-PCC-M: Principles of Compiler Construction.....	111
SWT-PR2-B: SWT Bachelorprojekt Software Systems Science.....	113
SWT-RSD-B: Reactive Systems Design.....	116
SWT-SEM-B: Seminar Software Engineering and Programming Languages (Bachelor).....	119
SWT-SSP-B: Soft Skills in IT-Projekten.....	121
SWT-SWL-B: Software Engineering Lab.....	123
SoSySc-Thesis-B: Bachelorarbeit in Software Systems Science.....	125
Stat-B-01: Methoden der Statistik I.....	126
Stat-B-02: Methoden der Statistik II.....	127
WI-Seminar-B: Bachelorseminar aus der Fächergruppen Wirtschaftsinformatik.....	128
WiMa-B-01a: Wirtschaftsmathematik I.....	129

---

---

# Übersicht nach Modulgruppen

## 1) A1 Mathematische Grundlagen (Modulgruppe) ECTS: 27

Stat-B-01: Methoden der Statistik I (6 ECTS, WS, SS).....	126
Stat-B-02: Methoden der Statistik II (6 ECTS, WS, SS).....	127
WiMa-B-01a: Wirtschaftsmathematik I (3 ECTS, WS, SS).....	129
Gdl-Mfi-1: Mathematik für Informatik 1 (Aussagen- u. Prädikatenlogik) (6 ECTS, WS, jährlich).....	38
KTR-Mfi-2: Mathematik für Informatik 2 (Lineare Algebra) (6 ECTS, SS, jährlich).....	67

## 2) A2 Allgemeine Informatik Grundlagen (Modulgruppe) ECTS: 36

DSG-EiAPS-B: Einführung in Algorithmen, Programmierung und Software (6 ECTS, WS, jährlich).....	16
PSI-EiRBS-B: Einführung in Rechner- und Betriebssysteme (6 ECTS, SS, jährlich).....	92
Gdl-GTI-B: Grundlagen der Theoretischen Informatik (6 ECTS, SS, jährlich).....	32
AI-AuD-B: Algorithmen und Datenstrukturen (6 ECTS, SS, jährlich).....	9
SWT-FSA-B: Foundations of Software Analysis (6 ECTS, WS, jährlich).....	105
MOBI-DBS-B: Datenbanksysteme (6 ECTS, SS, jährlich).....	83

## 3) A3 Programmierung und Softwaretechnik (Studienbeginn ab WS15/16) (Modulgruppe) ECTS: 18 - 42

### a) A3P Pflichtbereich Grundlagen PuSWT (Pflichtbereich) ECTS: 18

Gdl-IFP-B: Introduction to Functional Programming (6 ECTS, WS, jährlich).....	34
SWT-FSE-B: Foundations of Software Engineering (6 ECTS, SS, jährlich).....	107
SWT-SWL-B: Software Engineering Lab (6 ECTS, WS, jährlich).....	123

### b) A3W Wahlpflichtbereich Schwerpunkt PuSWE (Pflichtbereich) ECTS: 0 - 24

DSG-AJP-B: Fortgeschrittene Java Programmierung (3 ECTS, SS, jährlich).....	14
DSG-JaP-B: Java Programmierung (3 ECTS, WS, jährlich).....	22
HCI-IS-B: Interaktive Systeme (6 ECTS, WS, jährlich).....	44
KInf-SemInf-M: Semantische Informationsverarbeitung (6 ECTS, WS, jährlich).....	59
MI-WebT-B: Web-Technologien (6 ECTS, SS, jährlich).....	80
SME-Phy-B: Physical Computing (6 ECTS, SS, jährlich).....	101
SWT-MBT-B: Modellbasiertes Testen (3 ECTS, WS, jährlich).....	109

---

SWT-PCC-M: Principles of Compiler Construction (6 ECTS, SS, jährlich).....	111
SWT-RSD-B: Reactive Systems Design (6 ECTS, SS, jährlich).....	116

## **4) A4 Komplexe und Verteilte Systeme (Modulgruppe) ECTS: 21 - 42**

### **a) A4P Pflichtbereich Grundlagen KuVS (Pflichtbereich) ECTS: 21**

DSG-PKS-B: Programmierung komplexer interagierender Systeme (3 ECTS, WS, jährlich).....	24
KTR-Datkomm-B: Datenkommunikation (6 ECTS, WS, jährlich).....	63
DSG-IDistrSys-B: Introduction to Distributed Systems (6 ECTS, SS, jährlich).....	19
MOBI-MSS-B: Mobility in Software Systems (6 ECTS, WS, jährlich).....	85

### **b) A4W Wahlpflichtbereich Komplexe und Verteilte Systeme (Wahlpflichtbereich) ECTS: 0 - 21**

AI-KI-B: Einführung in die Künstliche Intelligenz (6 ECTS, SS, jährlich).....	12
Gdl-MTL: Modal and Temporal Logic (6 ECTS, WS, jährlich).....	36
HCI-KS-B: Kooperative Systeme (6 ECTS, SS, jährlich).....	46
HCI-US-B: Ubiquitäre Systeme (6 ECTS, WS, jährlich).....	51
KInf-GeoInf-B: Geoinformationssysteme (6 ECTS, SS, jährlich).....	57
PSI-IntroSP-B: Introduction to Security and Privacy (6 ECTS, WS, jährlich).....	94

## **5) A5 Fachstudium Anwendungsfächer (Modulgruppe) ECTS: 12 - 20**

Anwendungsfächer mit Bezug zu Software Systems Science sind Wahlpflichtmodule anderer Fakultäten im Umfang von insgesamt 12 bis 20 ECTS-Punkten aus Bachelornebenfächern gemäß des Anhangs der Allgemeinen Prüfungsordnung für Bachelor- und Masterstudiengänge der Fakultäten Geistes- und Kulturwissenschaften, Humanwissenschaften und Sozial- und Wirtschaftswissenschaften an der Otto-Friedrich-Universität Bamberg (vgl. <http://www.uni-bamberg.de/?id=29722>). Module aus dem Nebenfach Angewandte Informatik sind nicht wählbar.

**WICHTIG:** Die Modulprüfungen einzelner Module eines gewählten Anwendungsfachs müssen inklusive aller erforderlichen zugehörigen Modulteilprüfungen gemäß der gültigen Bestimmungen des gewählten Anwendungsfachs abgelegt werden. Die von der anbietenden Fakultät in der Studien- und Fachprüfungsordnung und im jeweiligen Modulhandbuch festgelegten Regelungen zur Anmeldung und Anrechnung von Modulprüfungen einzelner Module und ihrer zugeordneten Lehrveranstaltungen sind bindend. Im Hinblick auf einzelne Anwendungsfächer gelten oft Zugangsvoraussetzungen, so sind zum Teil Vorpraktika erforderlich. Zu den Details können die Informationen des jeweiligen Anwendungsfachs und die Informationsangebote der dortigen Fachstudienberatung herangezogen werden, vgl: <http://www.uni-bamberg.de/?id=29722> . Mit der ersten Prüfungsanmeldung zu einem Modul verpflichtet sich der Studierende sowohl zur Beachtung der Zuordnungsregelungen von Lehrveranstaltungen und Modulteilprüfungen des gewählten Moduls als auch der Anerkennung der Prüfungsanmeldungs- und -anrechnungsregelungen des Moduls in dem gewählten Anwendungsfach.

---

**6) A6 Kontextstudium (Modulgruppe) ECTS: 12 - 16**

**a) Allgemeine Schlüsselqualifikationen (Teil-Modulgruppe) ECTS: 0 - 16**

EESYS-IITP-B: Internationales IT-Projektmanagement (6 ECTS, SS, jährlich).....	30
ISM-PT-B: Methoden der Präsentation, Gesprächsführung und Diskussion (3 ECTS, WS, jährlich).....	54
ISM-TA-B: Technikfolgenabschätzung/-bewertung (3 ECTS, SS, jährlich).....	55
SWT-SSP-B: Soft Skills in IT-Projekten (3 ECTS, SS, jährlich).....	121

**b) Wissenschaftliches Arbeiten (Teil-Modulgruppe) ECTS: 0 - 3**

MI-WAIAI-B: Einführung in das wissenschaftliche Arbeiten für Informatik und Angewandte Informatik (3 ECTS, SS, jährlich).....	78
---	----

**c) Fremdsprachen (Wahlpflichtbereich) ECTS: 0 - 16**

Module gemäß des aktuellen Angebots des Sprachenzentrums, insb. IT English.

**d) Philosophie / Ethik (Teil-Modulgruppe) ECTS: 0 - 16**

Wählbar sind Module der Philosophie, die dem Studium Generale zugeordnet sind, sowie das folgende Modul:

PSI-EDS-B: Ethics for the Digital Society (3 ECTS, WS, jährlich).....	90
---	----

**7) A7 Seminare und Projekte (Modulgruppe) ECTS: 18**

Es sind zwei Seminarmodule in den Themenbereichen A2-A4 mit jeweils 3 ECTS-Punkten sowie ein Projektmodul Software Systems Science mit 12 ECTS-Punkten zu absolvieren. Mindestens ein Seminarmodul sowie das Projektmodul muss der Fächergruppe Informatik entstammen.

Die Zulassung zur Modulprüfung setzt bei den Modulen dieser Modulgruppe eine regelmäßige Teilnahme an den zugehörigen Lehrveranstaltungen gem. § 9 Abs. 10 APO voraus.

**a) Seminare (Wahlpflichtbereich) ECTS: 6**

**aa) Seminar(e) in Informatik (Fach) ECTS: 3 - 6**

DSG-Sem-B: Bachelorseminar zur Praktischen Informatik (3 ECTS, WS, SS).....	28
Gdl-Sem-B: Bachelorseminar Grundlagen der Informatik (3 ECTS, jährlich nach Bedarf WS und SS).....	42
KTR-Sem-B: Bachelorseminar zu Kommunikationssystemen und Rechnernetzen (3 ECTS, jährlich nach Bedarf WS und SS).....	72
MOBI-SEM-B: Bachelor-Seminar Mobile Software Systems (3 ECTS, WS, jährlich).....	89
PSI-Sem-B: Seminar Security and Privacy Foundations (3 ECTS, SS, jährlich).....	99

---

SWT-SEM-B: Seminar Software Engineering and Programming Languages (Bachelor) (3 ECTS, WS, SS).....119

**bb) Seminar in Angewandter Informatik oder Wirtschaftsinformatik (Fach) ECTS: 0 - 3**

HCI-Sem-B: Bachelorseminar Mensch-Computer-Interaktion (3 ECTS, SS, jährlich)..... 49

KInf-Seminar-B: Bachelorseminar Kulturinformatik (3 ECTS, SS, jährlich)..... 61

KogSys-Sem-B: Bachelorseminar Kognitive Systeme (3 ECTS, WS, jährlich)..... 74

MI-Sem-B: Bachelorseminar zur Medieninformatik (3 ECTS, SS, jährlich)..... 76

SME-Sem-B: Bachelorseminar zu Smart Environments (3 ECTS, WS, jährlich)..... 103

WI-Seminar-B: Bachelorseminar aus der Fächergruppen Wirtschaftsinformatik (3 ECTS, WS, SS)..... 128

**b) Projekte (Wahlpflichtbereich)**

GdI-PR2-B: Bachelorprojekt Software Systems Science (12 ECTS, WS, SS)..... 40

DSG-Project-2-SoSySc-B: DSG Bachelorprojekt Software Systems Science (12 ECTS, WS, jährlich)..... 26

KTR-SSSProj-B: KTR Bachelorprojekt Software Systems Science (12 ECTS, WS, SS)..... 69

SWT-PR2-B: SWT Bachelorprojekt Software Systems Science (12 ECTS, WS, SS)..... 113

MOBI-PRS-B: Bachelor Project Mobile Software Systems (SoSySc) (12 ECTS, WS, SS)..... 87

PSI-SSSPProject-B: Software Systems Science Project: Security and Privacy (12 ECTS, WS, SS)..... 96

**8) A8 Bachelorarbeit (Modulgruppe) ECTS: 12**

SoSySc-Thesis-B: Bachelorarbeit in Software Systems Science (12 ECTS, WS, SS).....125

---

<b>Modul AI-AuD-B Algorithmen und Datenstrukturen</b> <i>Algorithms and Data Structures</i>		6 ECTS / 180 h 42 h Präsenzzeit 138 h Selbststudium
(seit SS20) Modulverantwortliche/r: Prof. Dr. Diedrich Wolter		
<b>Inhalte:</b> Grundlegende Algorithmen (insbesondere Suchen, Sortieren, elementare Graphalgorithmen) und Datenstrukturen (insbesondere Listen, Hashtabellen, Bäume, Graphen) werden vorgestellt und analysiert. Konzepte der Korrektheit, Komplexität und Algorithmenkonstruktion werden eingeführt.		
<b>Lernziele/Kompetenzen:</b> Das Modul vermittelt Kompetenzen, Datenstrukturen und Algorithmen im Hinblick auf konkrete Anforderungen auswählen zu können, sie analysieren und durch Implementierung in einem Programm umsetzen zu können. Daneben sollen grundlegende Kompetenzen im Bereich der Algorithmenkonstruktion erworben werden. Durch die Übung soll auch die Fähigkeit zur Bewältigung von Programmieraufgaben erweitert werden, sowie Teamarbeit geübt werden.		
<b>Sonstige Informationen:</b> Ein Studium der Informatik erfordert grundsätzlich, sich Inhalte parallel zur Veranstaltung praktisch und theoretisch zu erschließen (Programmierung, Formalisierung, Beweisführung). Eine aktive Teilnahme an den Übungen sowie die Bearbeitung von Übungsaufgaben ist deshalb essentiell für den Studienerfolg in diesem Modul. Der Arbeitsaufwand gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Vorlesung 21h (14 Wochen à 1,5 Stunden)</li> <li>• Vor- und Nachbereitung der Vorlesung (inkl. Recherche und Studium zusätzlicher Quellen): ca. 20h</li> <li>• semesterbegleitendes Üben und Bearbeiten von Übungsaufgaben und Teilleistungen: ca. 80h</li> <li>• Tutorium 21h (14 Wochen à 1,5 Stunden)</li> <li>• Klausur sowie Klausurvorbereitung basierend auf dem erarbeiteten Stoff: ca. 40h</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundlegende Kenntnisse in Informatik und Programmierung wie sie in Modul DSG-EiAPS-B vermittelt werden, sowie Grundkenntnisse in der Programmierung in Java aus Modul DSG-JaP-B werden dringend empfohlen. Basiskenntnisse der Mathematik werden vorausgesetzt, insbesondere mathematische Notation und elementare Beweisführung, z.B. aus Modul Gdl-Mfl-1.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b> ab dem 2.	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Algorithmen und Datenstrukturen</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Diedrich Wolter <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich <b>Inhalte:</b>	<b>2,00 SWS</b>

<p>Die Vorlesung betrachtet die zentralen Bereiche des Themengebietes</p> <p>Algorithmen und Datenstrukturen:</p> <ul style="list-style-type: none"> <li>• Komplexitätsbegriffe (insb. Laufzeitkomplexität, Speicherplatzkomplexität, O-Notation)</li> <li>• Korrektheit von Algorithmen</li> <li>• Listen (einfach/doppelt verkettet, Stack, Queue)</li> <li>• Hashverfahren</li> <li>• Bäume (Datenstruktur, Traversierung, Binär-, AVL-, Suchbäume, Heap)</li> <li>• Graphen (Datenstruktur, DFS-, BFS-, Dijkstra-Algorithmus, grundlegende graphentheoretische Konzepte)</li> <li>• Sortieren</li> <li>• Suche in Texten</li> <li>• geometrische und räumliche Datenstrukturen</li> <li>• Algorithmenkonstruktion</li> </ul> <hr/> <p><b>Literatur:</b></p> <p>Als begleitende Lektüre wird ein Standardlehrbuch über Algorithmen und Datenstrukturen empfohlen, zum Beispiel:</p> <ul style="list-style-type: none"> <li>• Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest und Clifford Stein. Introduction to Algorithms, 3. Aufl., MIT Press, 2009</li> <li>• Guter Saake und Kai-Uwe Sattler Algorithmen und Datenstrukturen: Eine Einführung mit JAVA, ISBN: 978-3864901362, 5. Aufl. 2013, 576 Seiten, dpunkt.lehrbuch</li> <li>• Thomas Ottmann und Peter Widmayer: Algorithmen und Datenstrukturen, ISBN: 978-3827428035, 5. Aufl. 2012, 800 Seiten, Spektrum, Akademischer Verlag</li> </ul>	
<p><b>2. Algorithmen und Datenstrukturen</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Sprache:</b> Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b></p> <p>In der Übung werden Vorlesungsinhalte vertieft und deren praktische Anwendung geübt. Insbesondere werden folgende Aspekte betrachtet:</p> <ul style="list-style-type: none"> <li>• Verständnis und Nutzung von Algorithmen</li> <li>• Aufwandsbestimmung für Algorithmen</li> <li>• Implementierung von Algorithmen und Datenstrukturen</li> <li>• abstrakte Datentypen sowie Nutzung von Bibliotheken</li> <li>• Anwendung von Prinzipien zur Algorithmenkonstruktion</li> </ul> <hr/> <p><b>Literatur:</b></p> <p>siehe Vorlesung; weitere Literaturempfehlungen werden in der Übung bekanntgegeben</p>	<p><b>2,00 SWS</b></p>

<p><b>Prüfung</b></p> <p>schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p>	
---	--

**Beschreibung:**

Gegenstand der Klausur sind alle Inhalte von Vorlesung und Übung (einschließlich Teilleistungen; siehe unten). Zusätzlich zur Prüfungsdauer wird eine Lesezeit von 15 Minuten gewährt, um die zu bearbeitenden Aufgaben im Rahmen der Wahlmöglichkeiten auswählen zu können.

In der Klausur können 90 Punkte erzielt werden. Im Semester werden studienbegleitend Teilleistungen (schriftliche Hausarbeiten, typischerweise Programmieraufgaben) ausgegeben und besprochen, deren Abgabe freiwillig ist. Abgegebene Lösungsansätze werden bewertet; insgesamt sind maximal 12 Punkte zu erreichen. Bearbeitungsfristen werden in der ersten Veranstaltung bekanntgegeben. Ist die Klausur bestanden (in der Regel sind hierzu 50% der Punkte erforderlich), so werden die bei der Bearbeitung der Teilleistungen erreichten Punkte als Bonuspunkte angerechnet. Die Note 1,0 ist auch ohne Punkte aus Teilleistungen erreichbar.

<b>Modul AI-KI-B Einführung in die Künstliche Intelligenz</b> <i>Introduction to Artificial Intelligence</i>		6 ECTS / 180 h 42 h Präsenzzeit 138 h Selbststudium
(seit SS20) Modulverantwortliche/r: Prof. Dr. Diedrich Wolter Weitere Verantwortliche: Schmid, Ute, Prof. Dr.		
<b>Inhalte:</b> Dieses Modul bietet Studierenden einen Überblick über das Fachgebiet der Künstlichen Intelligenz (KI) und bietet eine Einführung in elementare Konzepte, Methoden und Algorithmen wie etwa Wissensrepräsentation, Suche, Wahrnehmung und Handlungsplanung. Die vermittelten Inhalte bilden eine Grundlage für kognitive und smarte Systeme sowie für wichtige Teile der Informatik und Wirtschaftsinformatik.  Themen: <ul style="list-style-type: none"> <li>• Problemstellungen im Fachgebiet KI</li> <li>• KI-Programmierung</li> <li>• intelligente Agenten</li> <li>• Wissensrepräsentation und Logik</li> <li>• Suche im Problemraum</li> <li>• maschinelles Lernen</li> <li>• Wahrnehmung</li> <li>• Unsicherheit</li> <li>• Handlungsplanung</li> </ul>		
<b>Lernziele/Kompetenzen:</b> <ul style="list-style-type: none"> <li>• Grundlegende Konzepte und Problemstellungen der KI definieren und erklären können</li> <li>• Einfache Ki-Algorithmen auf konkrete – auch neue – Problemstellungen anwenden können</li> <li>• Problemstellungen formal, insbesondere mit Mitteln der Logik modellieren können</li> <li>• Grundzüge von KI-Programmiertechniken (insbesondere funktionale und logische Programmierung) beherrschen</li> </ul>		
<b>Sonstige Informationen:</b> Die Vorlesung wird auf deutsch gehalten, die Folien sowie weitere Materialien sind überwiegend in englischer Sprache.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Fortgeschrittene Programmierkenntnisse (etwa durch Module DSG-EiAPS-B, DSG-JaP-B, Gdl-IFP erworben) sowie Kenntnisse von Basisalgorithmen (etwa durch das Modul AI-AuD-B, vormals MI-AuD-B) werden vorausgesetzt, ebenso die Bereitschaft, sich in neue Programmiersprachen und -paradigmen einzuarbeiten. Grundlegende Kenntnisse in Mathematik (insbesondere formale Notation und Beweisführung, z.B. erworben in Gdl-Mfl-1) sowie theoretischer Informatik (z.B. erworben in Gdl-GTI-B) werden empfohlen.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>

	1 Semester
<b>Lehrveranstaltungen</b>	
<b>1. Einführung in Künstliche Intelligenz</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Ute Schmid, Prof. Dr. Diedrich Wolter <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Lernziele:</b> siehe Modulbeschreibung	
<b>Inhalte:</b> Präsentation und Diskussion der Inhalte (siehe Modulbeschreibung), insbesondere theoretische und konzeptionelle Aspekte.	
<b>Literatur:</b> Stuart Russel und Peter Norvig (2010, 3. Auflage). Artificial Intelligence, a modern approach. Prentice Hall	
<b>2. Einführung Künstliche Intelligenz</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Lernziele:</b> siehe Modulbeschreibung	
<b>Inhalte:</b> praktische Vertiefungen zu den Inhalten der Vorlesung (siehe Modulbeschreibung)	
<b>Prüfung</b> schriftliche Modulprüfung (Klausur) / Prüfungsdauer: 90 Minuten <b>Beschreibung:</b> Schriftliche Prüfung zu Inhalten der Vorlesung und Übung im Umfang von 90 Minuten. Zugelassene Hilfsmittel werden in der ersten Lehrveranstaltung bekanntgegeben.  Zusätzlich zur Prüfungsdauer wird eine Lesezeit von 15 Minuten gewährt, um die zu bearbeitenden Aufgaben im Rahmen der Wahlmöglichkeiten auswählen zu können.  Im Semester werden studienbegleitend Teilleistungen in der Übung ausgegeben und besprochen, deren Abgabe freiwillig ist. Ist die Klausur bestanden, so werden die bei der Bearbeitung der Teilleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist dabei auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar. Die Anzahl der erreichbaren Bonuspunkte wird in der ersten Lehrveranstaltung bekanntgegeben.	

<b>Modul DSG-AJP-B Fortgeschrittene Java Programmierung</b> <i>Advanced Java Programming</i>		3 ECTS / 90 h 23 h Präsenzzeit 67 h Selbststudium
(seit WS16/17) Modulverantwortliche/r: Prof. Dr. Guido Wirtz		
<b>Inhalte:</b> Aufbauend auf den Grundkenntnissen der objekt-orientierten Programmierung in Java aus DSG-EiAPS-B soll der Umgang mit modernen objekt-orientierten Programmiersprachen durch einen genaueren Blick auf die Möglichkeiten, die eine moderne Programmierumgebung heute liefert, vertieft und gefestigt. Dazu gehören als Themen - jeweils am Beispiel 'Java' praktisch erläutert und geübt - insbesondere: <ul style="list-style-type: none"> <li>• Interfaces, abstrakte Klassen und komplexere Vererbungsstrukturen, Nutzung von Package-Strukturen,</li> <li>• Einsatz und Behandlung von Exceptions,</li> <li>• Nutzung komplexer Java-APIs, z.B. für Ein- und Ausgabe,</li> <li>• grundlegende XML Verarbeitung,</li> <li>• Debugging, Profiling und Testen,</li> <li>• Überblick über das Programmieren von (grafischen) Benutzerschnittstellen (G)UIs.</li> </ul> Zusätzlich werden die ersten Schritte zur Nutzung komplexer Programmierumgebungen, die über den einfachen Editor-Compiler-Ausführungs-Zyklus hinausgehen, insbesondere der Umgang mit einfachen Testszenarien zur Entwicklung verlässlicher Systeme, eingeübt.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden beherrschen die grundlegenden Mechanismen der objekt-orientierten Programmierung vertieft und sind auch in der Lage, einfache Probleme mit Hilfe der über die Standardprogrammiersprachen-Konstrukte hinausgehenden Hilfsmittel einer modernen Programmierumgebung effizient und flexibel zu lösen.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand von 90 Std. gliedert sich in <ul style="list-style-type: none"> <li>• 22.5 Std. Teilnahme an der Praktischen Übung</li> <li>• 55 Std. Bearbeiten der Programmieraufgabe (Assignment)</li> <li>• 12 Std. Vorbereitung auf das Abschlusskolloquium</li> <li>• 0.5 Std. Abschlusskolloquium inklusive Warten auf Ergebnis usw.</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Programmierkenntnisse in Java sowie Grundkenntnisse aus dem Bereich der Algorithmik und Softwareentwicklung, wie sie z.B. im Modul DSG-EiAPS-B vermittelt werden.  Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b> 2.	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Fortgeschrittene Java Programmierung</b> <b>Lehrformen:</b> kein Typ gewählt, Übung <b>Dozenten:</b> Prof. Dr. Guido Wirtz, Mitarbeiter Praktische Informatik <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Lernziele:</b> vgl. Modulbeschreibung	
<b>Inhalte:</b> vgl. Modulbeschreibung	
<b>Literatur:</b> Jedes weiterführende Buch zu Java ist verwendbar.	
<b>Prüfung</b> Hausarbeit mit Kolloquium / Prüfungsdauer: 10 Minuten Bearbeitungsfrist: 3 Monate <b>Beschreibung:</b> Die zu Beginn des Semesters ausgegebene Programmieraufgabe (Assignment) wird als Vorbereitung auf das Abschlusskolloquium von den Studierenden gelöst; zu den Lösungen gibt es inhaltliches Feedback und Hilfestellungen von den betreuenden Mitarbeitern schon während des Semesters. Im Abschlusskolloquium stellen die Studierenden (jeweils einzeln) die von ihnen während des Semesters erarbeiteten Lösungen zu der Programmieraufgabe vor, erläutern diese und beantworten Fragen zu ihrer Lösung und den dabei verwendeten Techniken und Sprachkonstrukten.	

<p><b>Modul DSG-EiAPS-B Einführung in Algorithmen, Programmierung und Software</b>  <i>Introduction to Algorithms, Programming and Software</i></p>	<p>6 ECTS / 180 h          45 h Präsenzzeit          135 h Selbststudium</p>
<p>(seit WS17/18 bis SS20)          Modulverantwortliche/r: Prof. Dr. Guido Wirtz</p>	
<p><b>Inhalte:</b></p> <p>Das Modul gibt einen ersten Einblick in die Informatik aus Sicht der Entwicklung von Algorithmen und deren Realisierung durch Programme in imperativen, objekt-orientierten und funktionalen Programmiersprachen (am Beispiel verschiedener Programmiersprachen und Programmierparadigmen) sowie einen Ausblick auf die Problematik der Softwareentwicklung. Behandelt werden die Grundprinzipien der Informatik zu:</p> <ul style="list-style-type: none"> <li>• Präsentation, Interpretation und Manipulation von Information,</li> <li>• Syntax und Semantik von einfachen Sprachen,</li> <li>• Probleme, Problemklassen und -Instanzen,</li> <li>• Design, Entwicklung und Implementierung von Algorithmen für einfache Problemklassen,</li> <li>• einfache Datenstrukturen wie Keller, Warteschlangen, Listen und Bäume,</li> <li>• Techniken zur Spezifikation, zur Datenabstraktion und funktionalen Abstraktion, z.B. Abstrakte Datentypen, sowie</li> <li>• einfache Beschreibungsmechanismen für Sprachen wie Grammatiken (Typ 2/3) und Automaten (Endliche Automaten, Kellerautomaten)</li> </ul> <p>All wichtigen Begriffe werden am Beispiel von konkreten Programmiersprachen veranschaulicht, so dass damit auch die Grundlagen imperativer, funktionaler sowie objektorientierter Programmiersprachen eingeführt werden. Dazu gehören insbesondere die für alle Programmiersprachen wesentlichen Konzepte wie</p> <ul style="list-style-type: none"> <li>• Wertebereiche, Namensräume und deren Realisierung durch Speichermodelle,</li> <li>• Seiteneffekte durch Zuweisungen vs. Berechnung von Funktionen (Parameter, Resultate),</li> <li>• Kontroll- und Datenfluss in einem Programm, bei Funktionsaufrufen usw.,</li> <li>• Iteration vs. Rekursion, sowie</li> <li>• Konzepte zur Strukturierung von Programmen.</li> </ul> <p>Neben der Diskussion der verschiedenen Konzepte werden auch die wichtigsten Aspekte durch praktisches Programmieren eingeübt.</p>	
<p><b>Lernziele/Kompetenzen:</b></p> <p>Studierende haben einen ersten Überblick über das Fach 'Informatik' mit seinen verschiedenen Gebieten und kennen die grundlegenden Begriffe, Methoden und Techniken der Informatik aus Sicht von Algorithmen, Programmiersprachen und Softwareentwicklung. Studierende sind in der Lage, aus einem Basisrepertoire an Möglichkeiten jeweils geeignete Abstraktions- und Repräsentationsmethoden zur maschinellen Bearbeitung auszuwählen und Methoden zur Beschreibung von Syntax und Semantik einfacher Sprachen anzuwenden. Sie verstehen die Zusammenhänge zwischen Spezifikation und Implementierung wie auch die grundsätzliche Arbeitsweise von Programmiersprachen und können die wesentlichen Schritte der Softwareentwicklung nachvollziehen. Studierende können einfache Problemstellungen beschreiben, algorithmische Lösungen dazu entwickeln und diese auch mittels einfacher Datenstrukturen in konkreten imperativen und funktionalen Programmiersprachen umsetzen.</p>	

**Sonstige Informationen:**

Auch wenn das Modul organisatorisch unabhängig von der Einführung in Java durch das ebenfalls jeweils im Wintersemester angebotene Modul DSG-JaP-B ist und beide Module auch bei entsprechenden Vorkenntnissen unabhängig voneinander absolviert werden können, **wird Studienanfängerinnen und -anfängern dringend empfohlen, beide Module im gleichen Semester zu bearbeiten, d.h. bei Studienbeginn zum Wintersemester im 1. Fachsemester und bei Studienbeginn zum Sommersemester jeweils im 2. Fachsemester.**

**Zulassungsvoraussetzung für die Belegung des Moduls:**

keine

**Empfohlene Vorkenntnisse:**

Die Veranstaltung hat als grundlegende Einführungsveranstaltung in das Gebiet der Informatik weder Inhalte anderer Lehrveranstaltungen noch Informatikkenntnisse oder Programmierkenntnisse zur Voraussetzung. *Insbesondere ist das Modul PSI-EiRBS-B (vormals DSG-EiRBS-B), das regelmäßig im Sommersemester angeboten wird, keine Voraussetzung für DSG-EiAPS-B.*

**Besondere****Bestehensvoraussetzungen:**

keine

**Angebotshäufigkeit:** WS, jährlich**Empfohlenes Fachsemester:**  
ab dem 1.**Minimale Dauer des Moduls:**  
1 Semester**Lehrveranstaltungen****1. Einführung in Algorithmen, Programmierung und Software****2,00 SWS****Lehrformen:** Vorlesung**Dozenten:** Prof. Dr. Guido Wirtz**Sprache:** Deutsch**Angebotshäufigkeit:** WS, jährlich**Lernziele:**

vgl. Modulbeschreibung

**Inhalte:**

vgl. Modulbeschreibung

**Literatur:**

Jede Einführung in die Informatik kann als Ergänzung zur Veranstaltung genutzt werden, allerdings orientiert sich die Vorlesung nicht an einem Buch; deshalb ist die Liste hier nur als Auswahl "nützlicher" Bücher zu verstehen, die zum Teil andere Schwerpunkte setzen, nicht unbedingt die gleichen Themen behandeln, bei gleichen Themen andere Herangehensweisen an das jeweilige Thema wählen und natürlich zum Teil andere Schreibweisen usw. benutzen:

- Heinz-Peter Gumm, Manfred Sommer: Einführung in die Informatik. Oldenbourg Verlag, 2013 (10th)
- Herbert Klaeren, Michael Sperber: Die Macht der Abstraktion - Einführung in die Programmierung. Teubner, 2007 (1th)
- Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi: How to Design Programs - An Introduction to Computing and Programming. The MIT Press 2001 (online verfügbar)

<ul style="list-style-type: none"> <li>• Helmut Herold, Bruno Lurz, Jürgen Wohlrab: Grundlagen der Informatik. Pearson Higher Education, 2012 (2nd)</li> <li>• Barbara Liskov with John Guttag: Program Development in Java. Addison-Wesley, 2001</li> </ul>	
<p><b>2. Einführung in Algorithmen, Programmierung und Software</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Mitarbeiter Praktische Informatik</p> <p><b>Sprache:</b> Deutsch</p> <p><b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Lernziele:</b></p> <p>vgl. Vorlesung</p> <hr/> <p><b>Inhalte:</b></p> <p>In der Übung werden die wichtigsten Konzepte der gleichnamigen Vorlesung an einfachen Beispielen praktisch umgesetzt und durch die Besprechung von typischen Aufgaben zum jeweiligen Thema, die den Studierenden regelmäßig zum freiwilligen Üben angeboten werden, vertieft. Dabei wird insbesondere Wert auf die Vorstellung von Lösungen durch die Studierenden und deren Diskussion in der Übungsgruppe gelegt.</p> <hr/> <p><b>Literatur:</b></p> <p>vgl. Vorlesung</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b></p> <p>schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b></p> <p>Klausur zum Stoff des gesamten Moduls, also der Vorlesung und Übung zur DSG-EiAPS-B. Bestehen der Klausur durch Erreichen von 50% der maximal erreichbaren Punkte.</p>	

<b>Modul DSG-IDistrSys-B Introduction to Distributed Systems</b>		6 ECTS / 180 h
<i>Introduction to Distributed Systems</i>		45 h Präsenzzeit
		135 h Selbststudium
(seit SS20)		
Modulverantwortliche/r: Prof. Dr. Guido Wirtz		
<b>Inhalte:</b>		
<p>Nowadays infrastructure and business relies more or less on distributed systems of various flavors. Most of our civilization would not work any more if all distributed systems would fail. So, that should be a good reason for anyone planning to work in the context of IT to learn at least about the characteristics and basic issues of such systems. The course introduces to the different flavors of and issues with distributed systems, discusses the most basic problems arising with this kind of systems and presents solutions and techniques that are essential to make distributed systems work. Additionally, the course also teaches how to build simple distributed systems using Java-based technologies like process interaction, synchronization, remote message invocation and web service infrastructure. Students are required to work (in groups) on an assignment using different technologies in order to combine the theoretical concepts with practical experience and ... Yes, we program!</p>		
<b>Lernziele/Kompetenzen:</b>		
<p>Students know about the characteristics and different flavors of distributed systems and understand the essential differences compared to monolithic, centralized systems as well as their consequences when designing and building distributed systems. Students are able to apply the basic algorithmic techniques and programming paradigms in order to build simple distributed systems themselves. Students have gained basic experience with practically building and running distributed systems.</p>		
<b>Sonstige Informationen:</b>		
<p>The language of instruction in this course is English.</p> <p>The overall workload of 180h for this module consists of:</p> <ul style="list-style-type: none"> <li>• weekly classes: 22.5h</li> <li>• tutorials: 22.5h</li> <li>• Work on assignment: 75h</li> <li>• Literature study 20h</li> <li>• preparation for and time of the final exam: 40h</li> </ul> <p>This course is intended for 2nd/3rd year bachelor students.</p>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
<p>Knowledge of the basics of computer science in general, esp. operating systems, as well as practical experience in Java programming, as the subjects taught in DSG-EiAPS-B and PSI-EiRBS. Preferable also knowledge about multithreading and synchronization like, e.g., the subject-matters of DSG-PKS-B.</p> <p>Modul Programmierung komplexer interagierender Systeme (DSG-PKS-B) - empfohlen</p>		keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>

	ab dem 4.	1 Semester
<b>Lehrveranstaltungen</b>		
<b>1. Introduction to Distributed Systems</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Guido Wirtz <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich		<b>2,00 SWS</b>
<b>Lernziele:</b> c.f. overall module description		
<b>Inhalte:</b> c.f. overall module description		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems - Concepts and Design. Pearson Education UK, 2011 (5. edition); ISBN: 9780273760597</li> <li>• Andrew Tanenbaum, Marten van Steen: Distributed Systems - Principles and Paradigms, 2017 (3rd edition)</li> </ul>		
<b>2. Tutorial Introduction to Distributed Systems</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Mitarbeiter Praktische Informatik <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich		<b>2,00 SWS</b>
<b>Lernziele:</b> c.f. overall module description		
<b>Inhalte:</b> Introduction to and discussion of tools and practical issues closely related to the topics discussed in the lecture as well as solutions of problems that come up during working on the practical assignment.		
<b>Prüfung</b> Hausarbeit mit Kolloquium / Prüfungsdauer: 10 Minuten Bearbeitungsfrist: 3 Monate <b>Beschreibung:</b> Oral examination concerning the topics discussed in the lecture, exercises and assignment. <b>Students may choose English or German as the language for the written assignment and oral examination.</b> Examinations will take place at the end of the summer term or at the begin of the winter term (students may choose one of them).  Students are assumed to work on a programming assignment ('schriftliche Hausarbeit') during the semester that is introduced at the beginning of the semester and uses the most important technologies discussed during the semester.		

---

**Note:** Without working on the programming assignment over the term students may run into problems during their oral examination (Kolloquium) as we discuss questions concerning topics from the lectures as well as from the assignment; questions about the assignment are based on the assignment solution programmed by the students.

<b>Modul DSG-JaP-B Java Programmierung</b> <i>Programming in Java</i>		3 ECTS / 90 h 30 h Präsenzzeit 60 h Selbststudium
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Guido Wirtz		
<b>Inhalte:</b> Die Programmiersprache Java wird als imperative und auch objektorientierte Sprache in ihren wichtigsten Konzepten und Bestandteilen eingeführt. Der Schwerpunkt liegt dabei auf dem Kern der Sprache und der praktischen Anwendung der Sprache zur Erstellung einfacher Programme, dem sog. Programmieren im Kleinen. Dazu werden die relevanten Konzepte zur Erstellung einfacher Programme (Schnittstellen, Klassen und Testklassen) zur Manipulation von grundlegenden Datentypen und einfachen Datenstrukturen (Felder, einfache Listen) eingeführt und anhand von Beispielen und durch einfache Programmieraufgaben erläutert. Zudem wird die Verwendung der in Java vorhandenen Techniken zur Umsetzung objektorientierten Programmierens wie z.B. Typisierung und Vererbung sowie Mechanismen zur Abstraktion und Strukturierung wie z.B. Schnittstellen und Paket-Strukturen und einfache Fehlerbehandlung (Exceptions) diskutiert.		
<b>Lernziele/Kompetenzen:</b> Studierende sollen mit der Programmiersprache Java und einer einfachen Programmierumgebung (Editor, Compiler, Dokumentations-Werkzeug) soweit praktisch umgehen können, dass sie aus der Beschreibung einfacher Probleme selbständig ein Lösungsverfahren entwickeln und durch Wahl geeigneter Daten- und Kontrollstrukturen in ein funktionsfähiges Java-Programm umsetzen, übersetzen und nach Auswahl geeigneter Testdaten testen können. Zusätzlich sollen die von Java angebotenen - für objektorientierte Sprachen typische - Strukturierungs- und Abstraktionstechniken wie Sichtbarkeitsbereiche, Schnittstellen und implementierende Klassen, einfache Vererbung, Programme bestehend aus mehreren Klassen sowie Verwendung eigener und vorgegebener Paketstrukturen sinnvoll eingesetzt werden können.		
<b>Sonstige Informationen:</b> Auch wenn das Modul organisatorisch unabhängig von der grundlegenden Einführung in die Informatik durch das Modul DSG-EiAPS-B ist und bei entsprechenden Vorkenntnissen auch ohne die DSG-EiAPS-B absolviert werden kann, wird Studienanfängerinnen und -anfängern <b>dringend empfohlen, beide Module im gleichen Semester zu bearbeiten</b> , d.h. bei Studienbeginn zum Wintersemester im 1. Fachsemester und bei Studienbeginn zum Sommersemester jeweils im 2. Fachsemester.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundlegende Kenntnisse in einfachen Algorithmen und Kontrollstrukturen sowie den Grundprinzipien der Informatik, wie sie z.B. in den ersten Sitzungen der Vorlesung zum Modul DSG-EiAPS-B vermittelt werden, werden vorausgesetzt.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>Java Programmierung</b> <b>Lehrformen:</b> Vorlesung, Übung		<b>2,00 SWS</b>

**Dozenten:** Prof. Dr. Guido Wirtz

**Sprache:** Deutsch

**Angebotshäufigkeit:** WS, jährlich

**Lernziele:**

Studierende sollen mit der Programmiersprache Java und einer einfachen Programmierumgebung (Editor, Compiler, Dokumentations-Werkzeug) soweit praktisch umgehen können, dass sie aus der Beschreibung einfacher Probleme selbständig ein Lösungsverfahren entwickeln und durch Wahl geeigneter Daten- und Kontrollstrukturen in ein funktionsfähiges Java-Programm umsetzen, übersetzen und nach Auswahl geeigneter Testdaten testen können. Zusätzlich sollen die von Java angebotenen - für objektorientierte Sprachen typische - Strukturierungs- und Abstraktionstechniken wie Sichtbarkeitsbereiche, Schnittstellen und implementierende Klassen, einfache Vererbung, Programme bestehend aus mehreren Klassen sowie Verwendung eigener und vorgegebener Paketstrukturen sinnvoll eingesetzt werden können.

**Inhalte:**

Die Programmiersprache Java wird als imperative und auch objektorientierte Sprache in ihren wichtigsten Konzepten und Bestandteilen eingeführt. Der Schwerpunkt liegt dabei auf dem Kern der Sprache und der praktischen Anwendung der Sprache zur Erstellung einfacher Programme, dem sog. Programmieren im Kleinen. Dazu werden die relevanten Konzepte zur Erstellung einfacher Programme (Schnittstellen, Klassen und Testklassen) zur Manipulation von grundlegenden Datentypen und einfachen Datenstrukturen (Felder, einfache Listen) eingeführt und anhand von Beispielen und durch einfache Programmieraufgaben erläutert. Zudem wird die Verwendung der in Java vorhandenen Techniken zur Umsetzung objektorientierten Programmierens wie z.B. Typisierung und Vererbung sowie Mechanismen zur Abstraktion und Strukturierung wie z.B. Schnittstellen und Paket-Strukturen und einfache Fehlerbehandlung (Exceptions) diskutiert.

**Prüfung**

schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten

**Beschreibung:**

Klausur zum Stoff des gesamten Moduls, insbesondere in Form von Programmieraufgaben. Bestehen der Klausur durch Erreichen von 50% der maximal erreichbaren Punkte.

<b>Modul DSG-PKS-B Programmierung komplexer interagierender Systeme</b> <i>Introduction to Parallel and Distributed Programming</i>		3 ECTS / 90 h 23 h Präsenzzeit 67 h Selbststudium
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Guido Wirtz		
<b>Inhalte:</b> Die Veranstaltung erläutert und übt den Umgang mit (explizit) parallelen Programmen und schafft damit auch ein vertieftes Verständnis für die Arbeitsweise heutiger Mehrkernprozessoren und Multiprozessoren. Dabei wird sowohl auf die grundlegenden Probleme und Techniken eingegangen als auch das praktische Entwerfen und Programmieren solcher Systeme (derzeit auf der Grundlage von Java) eingeübt. Dabei geht es um <ul style="list-style-type: none"> <li>• Prozesse und Threads,</li> <li>• Prozesskommunikation,</li> <li>• Synchronisation bei Shared Memory,</li> <li>• einfache C/S-Systeme mit TCP sockets,</li> <li>• Message-Passing im Aktor-Modell.</li> </ul> Zusätzlich wird die Problematik robuster verteilter Systeme diskutiert und ein Ausblick auf alternative Interaktionsparadigmen gegeben.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden kennen die gebräuchlichen Prozessbegriffe, die grundsätzlichen Probleme der Programmierung echt- und pseudo-paralleler Prozesssysteme sowie die grundlegenden Mechanismen zur Inter-Prozess-Kommunikation. Die Studierenden sind in der Lage, einfache parallele Programme mittels Threads zu schreiben, diese über Synchronisationsverfahren zu koordinieren sowie durch Kommunikationsmechanismen kooperativ und verlässlich zusammen arbeiten zu lassen.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand von 90 Std. gliedert sich in <ul style="list-style-type: none"> <li>• 22.5 Std. Teilnahme an der Praktischen Übung</li> <li>• 55 Std. Bearbeiten der Programmieraufgabe (Assignment)</li> <li>• 12 Std. Vorbereitung auf das Abschlusskolloquium</li> <li>• 0.5 Std. Abschlusskolloquium inklusive Warten auf Ergebnis usw.</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Programmierkenntnisse in Java sowie Grundkenntnisse aus dem Bereich der Betriebssysteme, wie sie z.B. im Modul PSI-EiRBS-B (vormals DSG-EiRBS-B) vermittelt werden.  Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Einführung in Rechner- und Betriebssysteme (PSI-EiRBS-B, vormals DSG-EiRBS-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<p><b>Programmierung komplexer interagierender Systeme</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Mitarbeiter Praktische Informatik</p> <p><b>Sprache:</b> Deutsch</p> <p><b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Lernziele:</b></p> <p>vgl. Modulbeschreibung</p> <hr/> <p><b>Inhalte:</b></p> <p>vgl. Modulbeschreibung</p> <hr/> <p><b>Literatur:</b></p> <p>- wird jeweils aktuell zur Veranstaltung angegeben -</p>	<b>2,00 SWS</b>
<p><b>Prüfung</b></p> <p>Hausarbeit mit Kolloquium / Prüfungsdauer: 10 Minuten</p> <p>Bearbeitungsfrist: 3 Monate</p> <p><b>Beschreibung:</b></p> <p>Die zu Beginn des Semesters ausgegebene Programmieraufgabe (Assignment) wird als Vorbereitung auf das Abschlusskolloquium von den Studierenden gelöst; zu den Lösungen gibt es inhaltliches Feedback und Hilfestellungen von den betreuenden Mitarbeitern schon während des Semesters. Im Abschlusskolloquium stellen die Studierenden (jeweils einzeln) die von ihnen während des Semesters erarbeiteten Lösungen zu der Programmieraufgabe vor, erläutern diese und beantworten Fragen zu ihrer Lösung und den dabei verwendeten Techniken und Sprachkonstrukten.</p>	

<p><b>Modul DSG-Project-2-SoSySc-B DSG Bachelorprojekt Software Systems Science</b>  <i>DSG Bachelorproject Software Systems Science</i></p>	<p>12 ECTS / 360 h                  220 h Präsenzzeit                  140 h Selbststudium</p>
<p>(seit WS18/19)                  Modulverantwortliche/r: Prof. Dr. Guido Wirtz</p>	
<p><b>Inhalte:</b>                  Überschaubare Themen aus der aktuellen Forschungsarbeit der Arbeitsgruppe Verteilte Systeme (DSG), die aber eine umfangreiche Einarbeitung erfordern können, werden in einer zum Teil gemeinsam, zum Teil arbeitsteilig, arbeitenden Gruppe von Studierenden von der Konzeption bis zur praktischen Umsetzung im Rahmen des zweisemestrigen Projekts durchgeführt. Dabei geht es nicht nur um die programmiertechnische Umsetzung, sondern insbesondere auch um die Entwicklung tragfähiger und mit den vorgegebenen Rahmenbedingungen kompatibler Konzepte zur Lösung der gestellten Aufgabe, sowie um die Sicherstellung der robusten und verlässlichen Funktion der entwickelten Systeme. In der Regel wird dazu das Studium aktueller Literatur und die Auswahl, Umsetzung und/oder Adaption zum Thema vorgeschlagener Ansätze notwendig sein. Typische Themen - die sich jeweils den aktuellen Arbeiten der DSG anpassen - sind z.B. die Untersuchung von BPMN- oder BPEL basierten Standards und Ansätzen im Bereich von dienst-orientierten Systemen oder aber die Erstellung eines Prototyps zum Monitoring oder der Visualisierung verteilter Software-Systeme.</p>	
<p><b>Lernziele/Kompetenzen:</b>                  Studierende sollen ein vertieftes Verständnis der bei der Durchführung von praktischen, arbeitsteilig organisierten, Softwareprojekten auftretenden Probleme wie auch von erfolgversprechenden Lösungsansätzen zu diesen Problemen erhalten. Da dies anhand der intensiven Bearbeitung eines Themas aus dem Forschungsbereich der Verteilten Systeme geschieht, gewinnen die TeilnehmerInnen wichtige Erfahrungen mit der Durchführung kleinerer, forschungsorientierter Projekte von der Grobkonzeption über die Detailplanung bis hin zur Umsetzung und Dokumentation der Ergebnisse in einem wissenschaftlich ausgerichteten Arbeitsbericht sowie einer Posterpräsentation.</p>	
<p><b>Sonstige Informationen:</b>                  Dieses Modul erstreckt sich über 2 Semester (Start im Wintersemester):                  2x6=12 ECTS, 2x4=8 SWS.                  Der Arbeitsaufwand beträgt insgesamt 360 Std., welche sich grob wie folgt gliedern:</p> <ul style="list-style-type: none"> <li>• 60 Std. Recherche, Planung und Teilnahme am Planungsworkshop</li> <li>• 40 Std. Teilnahme an Projekttreffen, einschließlich Tutorien</li> <li>• 180 Std. Durchführung des Projekts (Projektarbeit)</li> <li>• 20 Std. Erstellung des Zwischenberichts (Hausarbeit)</li> <li>• 60 Std. Erstellung des Abschlussberichts, sowie Erstellung und Präsentation des Projektposters (Hausarbeit und Kolloquium)</li> </ul>	
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>                  keine</p>	
<p><b>Empfohlene Vorkenntnisse:</b>                  Grundlegende methodische Kenntnisse zur Planung und Durchführung von Softwareprojekten und zum wissenschaftlichen Arbeiten, sowie Grundkenntnisse in der Programming paralleler und verteilter Systeme</p>	<p><b>Besondere Bestehensvoraussetzungen:</b>                  keine</p>

wie sie z.B. durch DSG-PKS-B und/oder DSG-IDistrSys vermittelt werden.		
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 2 Semester

<b>Lehrveranstaltungen</b>	
<p><b>DSG Bachelorprojekt Software Systems Science</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Prof. Dr. Guido Wirtz, Mitarbeiter Praktische Informatik</p> <p><b>Sprache:</b> Deutsch/Englisch</p> <p><b>Angebotshäufigkeit:</b> WS, SS</p> <hr/> <p><b>Lernziele:</b> siehe Modulbeschreibung</p> <hr/> <p><b>Inhalte:</b> vgl. Modulbeschreibung</p> <hr/> <p><b>Literatur:</b> Je nach Projektthematik; wird zu Beginn des Projekts bekannt gegeben.</p>	<b>8,00 SWS</b>

<p><b>Prüfung</b></p> <p>schriftliche Hausarbeit / Bearbeitungsfrist: 4 Monate</p> <p><b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regelmäßige Teilnahme an der Lehrveranstaltung</p> <p><b>Beschreibung:</b> Anfertigen eines schriftlichen Zwischenberichts zum Projekt in Deutsch oder Englisch nach etwa 80 Std. geleisteter Projektarbeit, spätestens am Ende des Semesters, in dem das Projekt begonnen wurde.</p> <p>Die Gewichtung der Prüfungsleistungen wird zu Beginn des Semesters bekannt gegeben.</p>	
---	--

<p><b>Prüfung</b></p> <p>Hausarbeit mit Kolloquium / Prüfungsdauer: 10 Minuten</p> <p>Bearbeitungsfrist: 4 Monate</p> <p><b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regelmäßige Teilnahme an der Lehrveranstaltung</p> <p><b>Beschreibung:</b> <b>Kolloquium:</b> Fachliche Diskussion auf der Grundlage des im Projekt bearbeiteten Themas im Rahmen einer Abschlussveranstaltung, auf der zunächst das zum Projekt angefertigte Poster erläutert wird; hier können auch praktische Projektergebnisse (z. B. lauffähige Software) demonstriert werden.</p> <p><b>Hausarbeit:</b> Anfertigen eines schriftlichen Abschlussberichts in Deutsch oder Englisch nach abgeschlossener Projektarbeit.</p> <p>Die Gewichtung der Prüfungsleistungen wird zu Beginn des Semesters bekannt gegeben.</p>	
--	--

<b>Modul DSG-Sem-B Bachelorseminar zur Praktischen Informatik</b>		3 ECTS / 90 h
<i>Bachelor Seminar in Practical Computer Science</i>		
(seit SS20)		
Modulverantwortliche/r: Prof. Dr. Guido Wirtz		
<b>Inhalte:</b> Verschiedene Themen aus dem Bereich der praktischen Informatik, die einen der fachlichen oder methodischen Aspekte aus den grundlegenden Informatik-Modulen DSG-EiAPS-B oder PSI-EiRBS-B anhand aktueller Literatur vertiefen und/oder ergänzen.		
<b>Lernziele/Kompetenzen:</b> Studierende sollen überschaubare aktuelle Themen der praktischen Informatik anhand eigener Literaturrecherchen unter Anleitung erarbeiten und in einer dem Thema angemessenen und für alle SeminarteilnehmerInnen verständlichen Form aufbereiten und präsentieren können.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundlegende Kenntnisse im jeweils im Seminar behandelten Gebiet der Praktischen Informatik, also mindestens eines der beiden Module DSG-EiAPS-B oder PSI-EiRBS-B.  Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Einführung in Rechner- und Betriebssysteme (PSI-EiRBS-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b> 2.	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorseminar zur Praktischen Informatik</b>	<b>2,00 SWS</b>
<b>Lehrformen:</b> Proseminar	
<b>Dozenten:</b> Prof. Dr. Guido Wirtz, Mitarbeiter Praktische Informatik	
<b>Sprache:</b> Deutsch	
<b>Angebotshäufigkeit:</b> WS, SS	
<b>Inhalte:</b> vgl. Modulbeschreibung	
<b>Literatur:</b> - wird jeweils nach Seminarthemen vergeben -	
<b>Prüfung</b> Hausarbeit mit Referat / Prüfungsdauer: 20 Minuten Bearbeitungsfrist: 4 Monate	
<b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regelmäßige Teilnahme an der Lehrveranstaltung	
<b>Beschreibung:</b>	

Begutachtung einer schriftlichen Ausarbeitung zu den wichtigsten Aspekten des erarbeiteten Themas mit formgerechter Liste der verwendeten Literatur.

Teilnahme am Peer-Review der Ausarbeitungen anderer Teilnehmer\*innen;

Freies Halten eines Referats auf der Grundlage der von dem/der Vortragenden erstellten Folien oder elektronischen Präsentationsunterlagen inklusive Diskussion der Inhalte mit den Seminarteilnehmerinnen und Seminarteilnehmern.

<b>Modul EESYS-IITP-B Internationales IT-Projektmanagement</b>		6 ECTS / 180 h
<i>International IT Project Management</i>		
(seit WS19/20)		
Modulverantwortliche/r: Prof. Dr. Thorsten Staake		
<b>Inhalte:</b>		
<p>Der Kurs vermittelt Fertigkeiten zum Management von und zur Mitarbeit in internationalen IT-Projekten. Behandelt werden alle Stufen entlang des Projektlebenszyklus von der initialen Betrachtung über die Planung, die Ausführung, das Projekt-Controlling und den Projektabschluss. Dabei finden sowohl traditionelle Vorgehensmodelle (z.B. Wasserfall- und V-Modelle) als auch agile Methoden (Scrum) Berücksichtigung. Behandelt werden ebenfalls Besonderheiten internationaler Teams und räumlich verteilter Projekte.</p> <p>Der Kurs verbindet die Vermittlung von Wissen zu Werkzeugen, etablierten Lösungstechniken, Bewertungsschemata etc. mit theoretischen Grundlagen und einer kritischen Auseinandersetzung bzgl. der Stärken und Grenzen der Ansätze.</p>		
<b>Lernziele/Kompetenzen:</b>		
Studierende sollen nach erfolgreichem Abschluss der Veranstaltung in der Lage sein, IT-Projekte in kleinen und großen Organisationen zu initiieren, zu planen, zu leiten und zu überwachen. Dazu können sie aus den behandelten Methoden und Vorgehensmodellen geeignete auswählen, die Wahl begründen, die Ansätze an neue und bisher unbekannte Problemstellungen anpassen und basierend auf den theoretischen Grundlagen fundiert und adäquat weiterentwickeln.		
<b>Sonstige Informationen:</b>		
Die Vorlesung findet als Hörsaalveranstaltung statt, bei der jedoch drei bis vier Vorlesungs-Einheiten ausschließlich als videobasierte Online-Veranstaltung durchgeführt werden. Die Online-Veranstaltungen umfassen Lehrvideos mit Untertiteln, weiterführende Literatur und online zu lösende Aufgaben.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
keine		keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		1 Semester
<b>Lehrveranstaltungen</b>		
<b>1. Internationales IT-Projektmanagement</b>		<b>2,00 SWS</b>
<b>Lehrformen:</b> Vorlesung		
<b>Dozenten:</b> Prof. Dr. Thorsten Staake		
<b>Sprache:</b> Deutsch/Englisch		
<b>Angebotshäufigkeit:</b> SS, jährlich		
<b>Inhalte:</b>		
Die Vorlesung behandelt die im Abschnitt „Modul EESYS-IITP-B“ unter „Inhalte“ genannten Themen. Die Erarbeitung der Kompetenzen wird durch Lehrvorträge, Fallstudien und Diskussionen unterstützt. Methoden und Konzepte werden regelmäßig anhand praktischer Beispiele eingeführt und in begrenztem		

<p>Umfang in Beispielaufgaben angewendet. Für einzelne Themen enthält die Vorlesung „Flipped-Classroom-Elemente“, bei denen erwartet wird, dass sich die Studierenden mit dem Lesen von Fachbeiträgen auf eine Veranstaltung vorbereiten, in der dann die Inhalte reflektiert und erweitert werden.</p>	
<p><b>Literatur:</b> Weiterführende Unterlagen werden in der Veranstaltung bekanntgegeben.</p>	
<p><b>2. Internationales IT-Projektmanagement</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b> In der Übung werden die in der Vorlesung behandelten Inhalte auf exemplarische Praxisprobleme angewendet, auf neue Fragestellungen übertragen und kritisch diskutiert. Besonderen Raum nehmen kleinere Fallstudien und die Analyse von Fachbeiträgen ein. Die Bearbeitung erfolgt in Teilen in Einzelarbeit und in Teilen in Kleingruppen. Die Übung transportiert auch vereinzelt neue Inhalte, insbesondere, wenn eine enge Verknüpfung mit deren Anwendung didaktisch sinnvoll ist. In einzelnen Übungen findet eine freiwillige, selbst zu korrigierende Lernfortschrittskontrolle statt.</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten <b>Beschreibung:</b> Durch die freiwillige Abgabe von semesterbegleitenden Studienleistungen können Punkte zur Notenverbesserung gesammelt werden, die auf die Klausur anrechenbar sind, sofern die Klausur auch ohne Punkte aus Studienleistungen bestanden ist. Zu Beginn der Lehrveranstaltung wird bekannt gegeben, ob Studienleistungen angeboten werden. Falls Studienleistungen angeboten werden, wird zu diesem Zeitpunkt auch die Anzahl, die Art, der Umfang und die Bearbeitungsdauer der Studienleistungen sowie die Anzahl an erreichbaren Punkten pro Studienleistung und in der Modulprüfung bekannt gegeben. Eine Bewertung von 1,0 kann auch ohne Punkte aus den Studienleistungen erreicht werden.</p>	

<b>Modul Gdl-GTI-B Grundlagen der Theoretischen Informatik</b> <i>Machines and Languages</i>		6 ECTS / 180 h
(seit WS18/19) Modulverantwortliche/r: Prof. Ph.D. Michael Mendler		
<b>Inhalte:</b> In der Veranstaltung wird die Theorie der Automaten, Sprachen und Algorithmen in ihren Grundzügen entwickelt. Das intuitiv einfach zu erfassende Modell der Turingmaschine als das Standardmodell der Berechenbarkeit und historischer Ausgangspunkt für die Entwicklung von programmierbaren Rechenmaschinen steht dabei im Mittelpunkt. Mit Turingmaschinen und anderer damit äquivalenter Berechnungsmodelle stößt die Veranstaltung zur Grenze dessen vor, was nach heutigem Wissen als prinzipiell maschinell berechenbar angesehen wird. Hierbei werden die wichtigsten Begriffe der Berechenbarkeits- und Komplexitätstheorie besprochen, insbesondere die Chomsky Hierarchie und Komplexitätsklassen wie P, NP, PSPACE, EXPTIME und ihre Beziehung untereinander.		
<b>Lernziele/Kompetenzen:</b> Kenntnis der wichtigsten Ergebnisse der Berechenbarkeits- und Komplexitätstheorie und den damit zusammenhängenden grundlegenden Einsichten in die Struktur und die Grenzen der Berechenbarkeit; Fähigkeit, Berechnungsmodelle unterschiedlicher Ausdruckskraft systematisch aufeinander zu reduzieren und die Turing-Äquivalenz von Programmiersprachen nachzuweisen oder zu widerlegen; Kenntnis konkreter mathematischer Grundmodelle zur Beschreibung von Algorithmus und Prozess, welche die wissenschaftlich-methodische Basis der Informatik bilden; Fähigkeit, rekursive und iterative Problemlösungen einerseits, sowie funktionale und reaktive Vorgänge andererseits gegeneinander abzugrenzen und ihre jeweilige Angemessenheit für die Modellierung praktischer Steuerungs- und Datenverarbeitungsaufgaben zu erkennen.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung und Übungen (inkl. Recherche und Studium zusätzlicher Quellen): 90 Stunden</li> <li>• Bearbeiten der Übungsaufgaben (unbenotet und freiwillig): 15 Stunden</li> <li>• Prüfungsvorbereitung + schriftliche Prüfung (90 min.): 30 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> gute Englischkenntnisse Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - Modul Mathematik für Informatik 1 (Aussagen- und Prädikatenlogik) (Gdl-MfI-1) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<p><b>1. Grundlagen der Theoretischen Informatik</b></p> <p><b>Lehrformen:</b> Vorlesung  <b>Dozenten:</b> Prof. Ph.D. Michael Mendler  <b>Sprache:</b> Deutsch/Englisch  <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b>  In der Vorlesung wird das Themengebiet der Veranstaltung durch Dozentenvortrag eingeführt und Anregungen zum weiterführenden Literaturstudium gegeben.  Die Lehrsprache wird in der ersten Lehrveranstaltung bekanntgegeben.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Hopcroft, J. E., Motwani, R., Ullman, J. D.: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, Pearson Studium, 2002.</li> <li>• Asteroth, A., Baier, Ch.: Theoretische Informatik, Pearson Studium, 2002.</li> <li>• Martin, J. C.: Introduction to Languages and the Theory of Computation, McGraw Hill, (2nd ed.), 1997.</li> </ul>	<b>2,00 SWS</b>
<p><b>2. Grundlagen der Theoretischen Informatik</b></p> <p><b>Lehrformen:</b> Übung  <b>Dozenten:</b> Prof. Ph.D. Michael Mendler, N.N.  <b>Sprache:</b> Englisch/Deutsch  <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b>  Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung an konkreten Beispielen. Sie dient damit auch der Klausurvorbereitung.  Die Lehrsprache wird in der ersten Lehrveranstaltung bekanntgegeben.</p> <hr/> <p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b>  Die Prüfungssprache ist Englisch.</p>	<b>2,00 SWS</b>

<b>Modul Gdl-IFP-B Introduction to Functional Programming</b> <i>Introduction to Functional Programming</i>		6 ECTS / 180 h
(seit WS19/20) Modulverantwortliche/r: Prof. Ph.D. Michael Mendler		
<b>Inhalte:</b> Die Veranstaltung beschäftigt sich mit den Grundlagen der funktionalen Programmierung als bedeutende Alternative zu herkömmlichen prozeduralen Sprachen. Diese nichtprozeduralen Sprachen, welche dem deklarativen und rekursiven Programmierprinzip folgen, werden besonders für ihre hohe Programmiereffizienz und -Sicherheit geschätzt. Der systematische Aufbau einer funktionalen Programmiersprache wird schrittweise erläutert und anhand konkreter Aufgabenstellungen nachvollzogen. Ausführliche praktische Übungen mit der Programmiersprache Haskell ergänzen die theoretischen Inhalte. Besonderes Augenmerk wird auf die Einführung in polymorphe Typsysteme gelegt und ihre Anwendung in der Typprüfung und Typsynthese als automatisches Softwarevalidierungsverfahren.		
<b>Lernziele/Kompetenzen:</b> Fähigkeit zur Entwicklung algorithmischer Problemlösungen in der funktionalen Programmiersprache Haskell; Kenntnis wichtiger Strukturprinzipien der funktionalen Programmierung, wie referenzielle Transparenz, Rekursion, induktive und coinduktive algebraische Datentypen, Monaden; Fähigkeit diese Strukturkonzepte adäquat in der Programmierung konkreter Aufgabenstellungen einzusetzen; Kenntnis deklarativer Modelle interaktiver Software und die Fähigkeit, diese in Haskell zu implementieren.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung und Übungen (inkl. Recherche und Studium zusätzlicher Quellen): 60 Stunden</li> <li>• Bearbeiten der Übungsaufgaben (unbenotet und freiwillig) und Teilnahme an Rechnerübungen: 45 Stunden</li> <li>• Prüfungsvorbereitung + schriftliche Prüfung (90 min.): 30 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> grundlegende Programmierkenntnisse, gute Englischkenntnisse Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (Gdl-Mfl-1) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>1. Introduction to Functional Programming</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Ph.D. Michael Mendler		<b>2,00 SWS</b>

<p><b>Sprache:</b> Englisch/Deutsch  <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b>  In der Vorlesung wird das Themengebiet der Veranstaltung durch Dozentenvortrag eingeführt und Anregungen zum weiterführenden Literaturstudium gegeben.  Die Lehrsprache wird in der ersten Lehrveranstaltung bekanntgegeben.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Pierce, B. C.: Types and Programming Languages, MIT Press, 2002</li> <li>• Thompson, S.: Haskell – The Craft of Functional Programming, Addison-Wesley 1999.</li> </ul>	
<p><b>2. Introduction to Functional Programming</b></p> <p><b>Lehrformen:</b> Übung  <b>Dozenten:</b> Prof. Ph.D. Michael Mendler  <b>Sprache:</b> Englisch/Deutsch  <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b>  Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung an konkreten Beispielen. Sie dient damit auch der Klausurvorbereitung.  Die Lehrsprache wird in der ersten Lehrveranstaltung bekanntgegeben.</p>	<b>2,00 SWS</b>
<p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b>  Die Prüfungssprache ist Englisch.</p>	

<b>Modul Gdl-MTL Modal and Temporal Logic</b> <i>Modal and Temporal Logic</i>		6 ECTS / 180 h
(seit WS17/18) Modulverantwortliche/r: Prof. Ph.D. Michael Mendler		
<b>Inhalte:</b> Nicht nur die Verifikation der funktionalen Korrektheit von Algorithmen und die funktionale Analyse verteilter und verlässlicher Systeme erfordert logisch-symbolische Verfahren. Auch viele Steuerungsprobleme in Anwendungsfeldern wie der Automatisierung von Wirtschaftsprozessen, intelligenten autonomen Agenten oder in Sicherheitsprotokollen lassen sich nur schwer mit herkömmlichen analytisch-numerischen Methoden behandeln. Dank der sich kontinuierlich verbessernden Leistungsfähigkeit moderner Rechner und der Erfolge im Gebiet der <i>Computational Logic</i> kommt der formalen Logik in der Informationstechnik wachsende Bedeutung zu. Die Vorlesung gibt einen Einblick in die Familie der Modallogiken als die wichtigsten informatikrelevanten Logiken, stellt zugehörige Implementierungstechniken und Entscheidungsverfahren vor und zeigt typische Anwendungen auf.		
<b>Lernziele/Kompetenzen:</b> Einsicht in die besondere Stellung der Modallogik zwischen Aussagenlogik und Prädikatenlogik und die Kenntnis ihrer ingenieurtechnischen Einsatzmöglichkeiten in Anwendungen, etwa der semantischen Informationsverarbeitung oder der Verifikation robuster und funktionssicherer reaktiver Systeme; Kenntnis der wichtigsten Modallogiken, ihrer Ausdruckskraft und Automatisierbarkeit, sowie die Fähigkeit für vorgegebene Anwendungen maßgeschneiderte Modallogiken selbst zu entwickeln; Fähigkeit, dynamische und reaktive Abläufe sowie komplexe verteilte Kommunikationsvorgänge in modaler und temporaler Logik zu spezifizieren und diese mit Hilfe geeigneter formaler Kalküle zu analysieren.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung und Übungen (inkl. Recherche und Studium zusätzlicher Quellen): 90 Stunden</li> <li>• Bearbeiten der Übungsaufgaben (unbenotet und freiwillig): 15 Stunden</li> <li>• Prüfungsvorbereitung: 30 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> grundlegende Programmierkenntnisse, gute Englischkenntnisse Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (Gdl-Mfl-1) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>Modal and Temporal Logic</b> <b>Lehrformen:</b> Vorlesung und Übung		<b>4,00 SWS</b>

<p><b>Dozenten:</b> Prof. Ph.D. Michael Mendler</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b></p> <p>In der Vorlesung wird das Themengebiet der Veranstaltung durch Dozentenvortrag eingeführt und Anregungen zum weiterführenden Literaturstudium gegeben.</p> <p>Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung an konkreten Beispielen. Sie dient damit auch der Klausurvorbereitung.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"><li>• Fagin, R., Halpern, J. Y., Moses, Y., Vardi, M. Y.: Reasoning about Knowledge. MIT Press, (2nd printing) 1996.</li><li>• Hughes, G. E., Cresswell, M. J.: A New Introduction to Modal Logic. Routledge, (3rd reprint) 2003.</li><li>• Popkorn, S.: First Steps in Modal Logic. Cambridge University Press, 1994.</li><li>• Berard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, Ph., McKenzie, P.: Systems and Software Verification. Springer 1999.</li></ul>	
--	--

<p><b>Prüfung</b></p> <p>schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b></p> <p>Die Prüfungssprache wird in der ersten Sitzung bekannt gegeben.</p>	
--	--

<b>Modul Gdl-Mfl-1 Mathematik für Informatik 1 (Aussagen- u. Prädikatenlogik)</b> <i>Propositional and Predicate Logic</i>		6 ECTS / 180 h
(seit SS17) Modulverantwortliche/r: Prof. Ph.D. Michael Mendler		
<b>Inhalte:</b> In dieser Basisvorlesung werden die für die Informatik wesentlichen Elemente der Aussagen- und Prädikatenlogik, sowie ihre Anwendung zur Spezifikation und Analyse diskreter Strukturen eingeführt. Am Beispiel der Prädikatenlogik wird der Prozess der Abstraktion im Aufbau und der Anwendung von formalen Systemen eingehend dargestellt. Der zentrale Unterschied zwischen Syntax und Semantik und das Prinzip rekursiver Konstruktionen und induktiven Schließens werden dabei ausführlich erläutert.		
<b>Lernziele/Kompetenzen:</b> Die Fähigkeit, informell gegebene Strukturen und Prozesse der natürlichen und technischen Umwelt, speziell solche mit nicht-numerischem Charakter mit symbolischen Formalismen zu erfassen und mit Hilfe kombinatorischer und logischer Lösungsansätze zu analysieren; Die Fähigkeit zur Abstraktion und die Einsicht in die methodische Bedeutung des hierarchischen Aufbaus informatischer Systeme, des systematischen Fortschreitens von einfachen zu komplexen Beschreibungen sowie umgekehrt des inkrementellen Abstützens komplexer Problemlösungen auf elementare Lösungsbausteine; Die Kenntnis elementarer Grundbegriffe der Beweis- und Modelltheorie der klassischen Aussagen- und Prädikatenlogik.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung und Übungen (inkl. Recherche und Studium zusätzlicher Quellen): 60 Stunden</li> <li>• Bearbeiten der Übungsaufgaben (unbenotet und freiwillig) und Teilnahme an Rechnerübungen: 45 Stunden</li> <li>• Prüfungsvorbereitung + schriftliche Prüfung (90 min.): 30 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> gute Englischkenntnisse		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Mathematik für Informatik 1</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Ph.D. Michael Mendler <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, jährlich <b>Inhalte:</b>	<b>2,00 SWS</b>

<p>In der Vorlesung wird das Themengebiet der Veranstaltung durch Dozentenvortrag eingeführt und Anregungen zum weiterführenden Literaturstudium gegeben.</p>	
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Ehrig, H., Mahr, B., Cornelius, F., Große-Rhode, Zeitz, M. P.: Mathematisch strukturelle Grundlagen der Informatik. Springer Verlag, 2. Aufl., 2001.</li> <li>• Grassmann, W. K., Tremblay, J.-P.: Logic and Discrete Mathematics - A Computer Science Perspective. Prentice Hall, 1996.</li> <li>• Scheinerman, E. R.: Mathematics – A Discrete Introduction. Brooks/Cole, 2000.</li> <li>• Barwise, J., Etchemendy, J: Language, Proof, and Logic. Seven Bridges Press, 2000.</li> </ul>	
<p><b>2. Mathematik für Informatik 1</b>  <b>Lehrformen:</b> Übung  <b>Dozenten:</b> Prof. Ph.D. Michael Mendler, N.N.  <b>Sprache:</b> Deutsch  <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b>  Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung an konkreten Beispielen. Sie dient damit auch der Klausurvorbereitung.</p>	<b>2,00 SWS</b>
<p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p>	

<p><b>Modul Gdl-PR2-B Bachelorprojekt Software Systems Science</b>  <i>Bachelor Project Software Systems Science</i></p>	<p>12 ECTS / 360 h</p>
<p>(seit WS18/19)                  Modulverantwortliche/r: Prof. Ph.D. Michael Mendler</p>	
<p><b>Inhalte:</b>                  Ein überschaubares Projekt zu Themen aus dem Bereich der aktuellen Forschungsarbeiten der Professur wird in arbeitsteilig arbeitenden Kleingruppen von Studierenden, ggf. auch einzeln - von der Konzeption und Umsetzung bis zur Dokumentation in einem wissenschaftlichen Arbeitsbericht und einer Präsentation - durchgeführt. Dies erfordert eine umfangreiche Recherche der aktuellen wissenschaftlichen Literatur sowie eine detaillierte Planung, deren Ergebnisse in einem Planungsworkshop vorgestellt und diskutiert werden.</p>	
<p><b>Lernziele/Kompetenzen:</b>                  Studierende sollen ein vertieftes Verständnis der bei der Durchführung von Softwaresystem-Projekten auftretenden konzeptionellen und praktischen Problem wie auch von geeigneten Lösungsansätzen erhalten. Dies geschieht in der intensiven Bearbeitung einer Aufgabenstellung mit Bezug zu den Grundlagen der Informatik in Kleingruppen - ggf. auch einzeln. Die Studenten sammeln wichtige Erfahrungen in der Durchführung kleinerer forschungsorientierter Projekte.</p>	
<p><b>Sonstige Informationen:</b>                  Dieses Modul soll über einen Zeitraum von 2 Semestern bearbeitet werden (2x6ECTS, 2x4SWS). In besonderen Fällen, z.B. Abwesenheit aufgrund eines Auslandsstudiums, kann das Projekt auch innerhalb einen Semesters plus Semesterferien durchgeführt werden. Studenten werden gebeten in solchen Fällen das Thema und den zeitlichen Ablauf schon im Semester vor Beginn des Projektes mit dem betreuenden Dozenten festzulegen.                   Der Arbeitsaufwand beträgt insgesamt 360 Std, welche sich grob wie folgt aufteilen:</p> <ul style="list-style-type: none"> <li>• 60 Std. Recherche, Planung und Teilnahme am Planungsworkshop</li> <li>• 40 Std. Teilnahme an Projekttreffen, einschließlich Tutorien</li> <li>• 180 Std. Durchführung des Projekts</li> <li>• 20 Std. Erstellung des Zwischenberichts (Hausarbeit)</li> <li>• 60 Std. Erstellung Abschlussbericht, sowie Erstellung und Präsentation des Projektposters (Hausarbeit und Kolloquium)</li> </ul> <p>Berichte und Präsentation dürfen wahlweise in Deutsch oder Englisch abgefasst sein.</p>	
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>                  keine</p>	
<p><b>Empfohlene Vorkenntnisse:</b>                  Grundlegende methodische Kenntnisse zur Planung und Durchführung von Softwareprojekten, z.B. erworben im Modul "Software Engineering Lab" und zum wissenschaftlichen Arbeiten, z.B. erworben im Modul "Wissenschaftliches Arbeiten in der Informatik".                   Darüber hinaus sind für Projekte im Bereich Gdl empfohlen:                  Englischkenntnisse, Kenntnisse der elementaren Logik, Grundlagen der Theoretischen Informatik, Rechner- und Betriebssysteme, Nichtprozedurale Programmierung.</p>	<p><b>Besondere Bestehensvoraussetzungen:</b>                  keine</p>

<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 2 Semester
-----------------------------------	----------------------------------	---

<b>Lehrveranstaltungen</b>	
<b>Gdl Bachelorprojekt Software Systems Science</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Prof. Ph.D. Michael Mender, N.N. <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, SS	<b>8,00 SWS</b>
<b>Inhalte:</b> Die Prüfungssprache wird in der ersten Lehrveranstaltung bekanntgegeben.	

<b>Prüfung</b> Hausarbeit mit Kolloquium / Prüfungsdauer: 30 Minuten Bearbeitungsfrist: 4 Monate <b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regelmäßige Teilnahme an der Lehrveranstaltung <b>Beschreibung:</b> Anfertigen eines schriftlichen Abschlussberichts und Kolloquium (Fachgespräch) zu den Ergebnisse des Projektes.  Die Gewichtung der Prüfungsleistungen wird zu Beginn der Lehrveranstaltung von der Dozentin bzw. dem Dozenten bekanntgegeben.  Die Prüfungssprache wird in der ersten Lehrveranstaltung bekanntgegeben.	
--	--

<b>Prüfung</b> schriftliche Hausarbeit / Bearbeitungsfrist: 4 Monate <b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regelmäßige Teilnahme an der Lehrveranstaltung <b>Beschreibung:</b> Anfertigen eines schriftlichen Zwischenberichts zum Projekt nach etwa 80 Std. geleisteter Projektarbeit, spätestens am Ende des Semesters, in dem das Projekt begonnen wurde.  Die Prüfungssprache sowie die Gewichtung der Prüfungsleistungen wird in der ersten Lehrveranstaltung bekanntgegeben.	
--	--

<b>Modul Gdl-Sem-B Bachelorseminar Grundlagen der Informatik</b>		3 ECTS / 90 h
<i>Seminar Foundations of Computing</i>		
(seit WS17/18)		
Modulverantwortliche/r: Prof. Ph.D. Michael Mendler		
<b>Inhalte:</b> Im Seminarmodul werden wechselnde Themen im Bereich der Informatikgrundlagen angeboten.		
<b>Lernziele/Kompetenzen:</b> Fähigkeit zur selbständigen Erarbeitung von Inhalten aus der aktuellen wissenschaftlichen Literatur; Fähigkeit, komplexe Problemlösungsansätze schriftlich und mündlich zu vermitteln. Förderung der wissenschaftliche Neugier und die Ausbildung einer selbstbewussten und forschenden Einstellung zur Technik.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Seminarvorträgen: 15 Stunden</li> <li>• Recherche und Literaturstudium: 25 Stunden</li> <li>• Vorbereitung des Seminarvortrags und schriftliche Ausarbeitung: 50 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Mathematik für Informatiker, Einführung in die Informatik, Rechner- und Betriebssysteme, Grundlagen der Theoretischen Informatik, gute Englischkenntnisse.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> jährlich nach Bedarf WS und SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Grundlagen der Informatik</b> <b>Lehrformen:</b> Seminar <b>Dozenten:</b> Michael Mendler, N.N. <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> jährlich nach Bedarf WS und SS	<b>2,00 SWS</b>
<b>Inhalte:</b> Das Gdl-Seminar wird zu semesterweise wechselnden Themen angeboten. Die Lehrsprache wird in der ersten Lehrveranstaltung bekanntgegeben.	
<b>Literatur:</b> Literatur wird bei Ankündigung bzw. zu Beginn des Seminars bekanntgegeben.	
<b>Prüfung</b> Hausarbeit mit Referat / Prüfungsdauer: 30 Minuten Bearbeitungsfrist: 4 Monate <b>Zulassungsvoraussetzung zur Modulprüfung:</b> regelmäßige Teilnahme an der Lehrveranstaltung <b>Beschreibung:</b>	

---

Die Prüfungssprache wird in der ersten Lehrveranstaltung bekanntgegeben.

<b>Modul HCI-IS-B Interaktive Systeme</b> <i>Interactive Systems</i>		6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Tom Gross		
<b>Inhalte:</b> Theoretische, methodische und praktische Grundlagen der Mensch-Computer-Interaktion.		
<b>Lernziele/Kompetenzen:</b> Ziel ist die allgemeine Einführung und Vermittlung von grundlegenden Paradigmen, Konzepten und Prinzipien der Gestaltung von Benutzungsoberflächen. Der primäre Fokus liegt dabei auf dem Entwurf, der Implementation und der Evaluierung von interaktiven Systemen.		
<b>Sonstige Informationen:</b> <a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: insgesamt 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung (inkl. Recherche und Studium zusätzlicher Quellen): ca. 30 Stunden</li> <li>• Vor- und Nachbereitung der Übung (inkl. Recherche und Studium zusätzlicher Quellen, aber ohne Bearbeitung der optionalen Studienleistungen): ca. 30 Stunden</li> <li>• Bearbeiten der optionalen Studienleistungen: insgesamt ca. 45 Stunden</li> <li>• Prüfungsvorbereitung: ca. 30 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li> </ul> Der Unterricht erfolgt grundsätzlich in deutscher und bei Bedarf der Studierenden in englischer Sprache. Sämtliche Unterlagen (inkl. Prüfung) sind in englischer Sprache verfügbar.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundkenntnisse in Informatik im Umfang einer Einführung in die Informatik		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Interaktive Systeme</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Tom Gross <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Im Rahmen der Vorlesung werden die folgenden Themen behandelt: <ul style="list-style-type: none"> <li>• Einführung in die Gestaltung von Benutzungsoberflächen</li> <li>• Benutzer und Humanfaktoren</li> <li>• Maschinen und technische Faktoren</li> <li>• Interaktion, Entwurf, Prototyping und Entwicklung</li> </ul>	

<ul style="list-style-type: none"> <li>• Evaluierung von interaktiven Systemen</li> <li>• Entwicklungsprozess interaktiver Systeme</li> <li>• Interaktive Systeme im größeren Kontext und verwandte Themen</li> </ul>	
<p><b>Literatur:</b> Die Veranstaltung ist eine Zusammenstellung verschiedener Quellen; als ergänzende Quelle und zum Nachschlagen wird empfohlen:</p> <ul style="list-style-type: none"> <li>• Preece, J., Rogers, Y. und Sharp, H. Interaction Design: Beyond Human-Computer Interaction. Wiley, New York, NY, 3. Auflage, 2011</li> <li>• Dix, A., Finlay, J., Abowd, G.D. und Beale, R. Human-Computer Interaction. Pearson, Englewood Cliffs, NJ, 3. Auflage, 2004.</li> </ul>	
<p><b>2. Interaktive Systeme</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Mitarbeiter Mensch-Computer-Interaktion <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b> praktische Aufgaben zum Vorlesungsstoff einschließlich der Programmierung kleiner Prototypen</p> <hr/> <p><b>Literatur:</b> siehe Vorlesung</p>	<b>2,00 SWS</b>
<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b> In der <b>Klausur</b> können 90 Punkte erzielt werden.</p> <p>Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen, die Punkte pro optionaler Studienleistung sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die Klausur bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der Klausur auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.</p>	

<b>Modul HCI-KS-B Kooperative Systeme</b> <i>Cooperative Systems</i>		6 ECTS / 180 h
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Tom Gross		
<b>Inhalte:</b> Theoretische, methodische und praktische Grundlagen der rechnergestützten Gruppenarbeit.		
<b>Lernziele/Kompetenzen:</b> Ziel ist die Vermittlung von grundlegenden Paradigmen und Konzepten von Rechnergestützter Gruppenarbeit (Computer-Supported Cooperative Work; CSCW) sowie die daraus resultierenden Designprinzipien und Prototypen. Dabei wird der Begriff breit gefasst; das zentrale Anliegen ist entsprechend die generelle technische Unterstützung von sozialer Interaktion, welche vom gemeinsamen Arbeiten und Lernen bis zum privaten Austausch reichen kann.		
<b>Sonstige Informationen:</b> <a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: insgesamt 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung (inkl. Recherche und Studium zusätzlicher Quellen): ca. 30 Stunden</li> <li>• Vor- und Nachbereitung der Übung (inkl. Recherche und Studium zusätzlicher Quellen, aber ohne Bearbeitung der optionalen Studienleistungen): ca. 30 Stunden</li> <li>• Bearbeiten der optionalen Studienleistungeng: insgesamt ca. 45 Stunden</li> <li>• Prüfungsvorbereitung: ca. 30 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li> </ul> Der Unterricht erfolgt grundsätzlich in deutscher und bei Bedarf der Studierenden in englischer Sprache. Sämtliche Unterlagen (inkl. Prüfung) sind in englischer Sprache verfügbar.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundkenntnisse in Informatik im Umfang einer Einführung in Algorithmen, Programmierung und Software, sowie Programmierkenntnisse in Java.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Kooperative Systeme</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Tom Gross <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Im Rahmen der Vorlesung werden nach einer Einführung in das Thema die folgenden Themen behandelt: <ul style="list-style-type: none"> <li>• Grundlegende Konzepte</li> </ul>	

<ul style="list-style-type: none"> <li>• Technologische Unterstützung für wechselseitige Information, Kommunikation, Koordination, Gruppenarbeit und Online-Gemeinschaften</li> <li>• Analyse kooperativer Umgebungen</li> <li>• Entwurf von CSCW und Groupware</li> <li>• Implementation von CSCW und Groupware</li> <li>• CSCW im größeren Kontext und verwandte Themen</li> </ul>	
<p><b>Literatur:</b> Die Veranstaltung ist eine Zusammenstellung verschiedener Quellen; als ergänzende Quelle und zum Nachschlagen wird empfohlen:</p> <ul style="list-style-type: none"> <li>• Gross, T. und Koch, M. Computer-Supported Cooperative Work. Oldenbourg, München, 2007.</li> <li>• Borghoff, U.M. und Schlichter, J.H. Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer-Verlag, Heidelberg, 2000.</li> </ul>	
<p><b>Prüfung</b> mündliche Prüfung / Prüfungsdauer: 30 Minuten</p> <p><b>Beschreibung:</b> In Abhängigkeit von der Teilnehmerzahl wird die Modulprüfung entweder in Form einer Klausur <b>oder</b> in Form einer mündlichen Prüfung durchgeführt. Die Festlegung erfolgt zu Semesterbeginn und wird im ersten Lehrveranstaltungstermin bekannt gegeben.</p> <p>In der mündlichen Prüfung mit einer Prüfungsdauer von 30 Minuten können 90 Punkte erzielt werden.</p> <p>Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen, die Punkte pro optionaler Studienleistung sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die mündl. Prüfung bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der mündl. Prüfung auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.</p>	
<p><b>Lehrveranstaltungen</b></p>	
<p><b>Kooperative Systeme</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Mitarbeiter Mensch-Computer-Interaktion <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b> praktische Aufgaben zum Vorlesungsstoff einschließlich der Programmierung kleiner Prototypen</p> <hr/> <p><b>Literatur:</b> siehe Vorlesung</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b></p>	

schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten

**Beschreibung:**

In Abhängigkeit von der Teilnehmerzahl wird die Modulprüfung entweder in Form einer Klausur **oder** in Form einer mündlichen Prüfung durchgeführt. Die Festlegung erfolgt zu Semesterbeginn und wird im ersten Lehrveranstaltungstermin bekannt gegeben.

In der Klausur über 90 min. können 90 Punkte erzielt werden.

Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen, die Punkte pro optionaler Studienleistung sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die Klausur bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der Klausur auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.

<b>Modul HCI-Sem-B Bachelorseminar Mensch-Computer-Interaktion</b>		3 ECTS / 90 h
<i>Bachelor-Seminar Human-Computer Interaction</i>		
(seit WS17/18)		
Modulverantwortliche/r: Prof. Dr. Tom Gross		
<b>Inhalte:</b> Aktive wissenschaftliche Bearbeitung aktueller Konzepte, Technologien und Werkzeuge der Mensch-Computer-Interaktion.		
<b>Lernziele/Kompetenzen:</b> Ziel ist das Erlernen des eigenständigen Erarbeitens und Präsentierens von Themengebieten aus dem Fach Mensch-Computer-Interaktion auf Basis der Literatur. Dabei werden die Fähigkeiten im Bereich der kritischen und systematischen Literaturbetrachtung ebenso weiterentwickelt wie die Kompetenzen in der Präsentation von Fachthemen.		
<b>Sonstige Informationen:</b> <a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a>  Der Arbeitsaufwand für dieses Modul gliedert sich typischerweise in folgende Bereiche: <ul style="list-style-type: none"> <li>• Teilnahme an den Präsenzveranstaltungen (Themenvergabe, Besprechungen, Präsentationen): ca. 20 Stunden</li> <li>• Literaturrecherche und Einarbeitung: ca. 25 Stunden</li> <li>• Vorbereitung der Präsentation: ca. 15 Stunden</li> <li>• Erstellen der schriftlichen Ausarbeitung: ca. 30 Stunden</li> </ul> Der Unterricht erfolgt grundsätzlich in deutscher und bei Bedarf der Studierenden in englischer Sprache. Sämtliche Unterlagen (inkl. Prüfung) sind in englischer Sprache verfügbar.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Modul Interaktive Systeme (HCI-IS-B)		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Mensch-Computer-Interaktion</b> <b>Lehrformen:</b> Seminar <b>Dozenten:</b> Prof. Dr. Tom Gross, Mitarbeiter Mensch-Computer-Interaktion <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Im Seminar werden aufbauend auf den in den Vorlesungen und Übungen des Faches Mensch-Computer-Interaktion erworbenen Kenntnissen und Fertigkeiten wechselnde aktuelle Forschungsthemen zu deren Inhalten bearbeitet. Dabei sind im Regelfall Aspekte mehrerer Lehrveranstaltungen relevant.	
<b>Literatur:</b>	

wird jeweils zu Beginn der Veranstaltung bekannt gegeben	
<b>Prüfung</b> Hausarbeit mit Referat / Prüfungsdauer: 30 Minuten Bearbeitungsfrist: 4 Monate <b>Beschreibung:</b> Schriftliche Ausarbeitung und Vortrag zu dem im Seminar von der Teilnehmerin bzw. vom Teilnehmer bearbeiteten Thema, inkl. Diskussion	

<b>Modul HCI-US-B Ubiquitäre Systeme</b> <i>Ubiquitous Systems</i>		6 ECTS / 180 h
(seit SS20) Modulverantwortliche/r: Prof. Dr. Tom Gross		
<b>Inhalte:</b> Theoretische, methodische und praktische Grundlagen des Ubiquitous Computing.		
<b>Lernziele/Kompetenzen:</b> Ziel ist die Vermittlung fundierter Kenntnisse und Fähigkeiten im Bereich der ubiquitären Systeme sowie eines breiten theoretischen und praktischen Methodenwissens zum Entwurf, zur Konzeption und zur Evaluierung ubiquitärer Systeme. Nach dem Besuch dieser Lehrveranstaltung sollen Studierende die einschlägige Literatur und Systeme in Breite und Tiefe kennen und neue Literatur und Systeme kritisch bewerten können.		
<b>Sonstige Informationen:</b> <a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: insgesamt 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung (inkl. Recherche und Studium zusätzlicher Quellen): ca. 30 Stunden</li> <li>• Vor- und Nachbereitung der Übung (inkl. Recherche und Studium zusätzlicher Quellen sowie Bearbeitung der optionalen Studienleistungen): ca. 75 Stunden</li> <li>• Prüfungsvorbereitung: ca. 30 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li> </ul> Der Unterricht erfolgt grundsätzlich in deutscher und bei Bedarf der Studierenden in englischer Sprache. Sämtliche Unterlagen (inkl. Prüfung) sind in englischer Sprache verfügbar.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Modul Algorithmen und Datenstrukturen (AI-AuD-B, vormals MI-AuD-B) Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B)		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Ubiquitäre Systeme</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Tom Gross <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Im Rahmen der Vorlesung werden nach einer Einführung in das Thema Ubiquitous Computing - also der allgegenwärtigen Rechner, die verschwindend klein, teilweise in Alltagsgegenständen eingebaut, als Client und Server fungieren	

<p>und miteinander kommunizieren können - die folgenden Themen konzeptionell, technisch und methodisch behandelt:</p> <ul style="list-style-type: none"> <li>• Grundlegende Konzepte</li> <li>• Basistechnologie und Infrastrukturen</li> <li>• Ubiquitäre Systeme und Prototypen</li> <li>• Kontextadaptivität</li> <li>• Benutzerinteraktion</li> <li>• Ubiquitäre Systeme im größeren Kontext und verwandte Themen</li> </ul> <hr/> <p><b>Literatur:</b> Die Veranstaltung ist eine Zusammenstellung verschiedener Quellen; als ergänzende Quelle und zum Nachschlagen wird empfohlen:</p> <ul style="list-style-type: none"> <li>• Krumm, J., (Hrsg.). Ubiquitous Computing Fundamentals. Taylor &amp; Francis Group, Boca Raton, FL, 2010.</li> </ul>	
<p><b>Prüfung</b> mündliche Prüfung / Prüfungsdauer: 30 Minuten</p> <p><b>Beschreibung:</b> In Abhängigkeit von der Teilnehmerzahl wird die Modulprüfung entweder in Form einer Klausur <b>oder</b> in Form einer mündlichen Prüfung durchgeführt. Die Festlegung erfolgt zu Semesterbeginn und wird im ersten Lehrveranstaltungstermin bekannt gegeben.</p> <p>In der mündlichen Prüfung mit einer Prüfungsdauer von 30 Minuten können 90 Punkte erzielt werden.</p> <p>Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen, die Punkte pro optionaler Studienleistung sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die mündl. Prüfung bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der mündl. Prüfung auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.</p>	
<p><b>Lehrveranstaltungen</b></p>	
<p><b>Ubiquitäre Systeme</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Mitarbeiter Mensch-Computer-Interaktion <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b> praktische Aufgaben zum Vorlesungsstoff einschließlich der Programmierung kleiner Prototypen.</p> <hr/> <p><b>Literatur:</b> siehe Vorlesung</p>	<p><b>2,00 SWS</b></p>

**Prüfung**

schriftliche Modulprüfung (Klausur) / Prüfungsdauer: 90 Minuten

**Beschreibung:**

In Abhängigkeit von der Teilnehmerzahl wird die Modulprüfung entweder in Form einer Klausur **oder** in Form einer mündlichen Prüfung durchgeführt. Die Festlegung erfolgt zu Semesterbeginn und wird im ersten Lehrveranstaltungstermin bekannt gegeben.

In der Klausur über 90 min. können 90 Punkte erzielt werden.

Es besteht die Möglichkeit, optionale Studienleistungen zu erbringen. Diese umfassen insgesamt 12 Punkte. Die Art der optionalen Studienleistungen, die Punkte pro optionaler Studienleistung sowie deren Bearbeitungsfrist werden zu Beginn der Lehrveranstaltung verbindlich bekannt gegeben. Ist die Klausur bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die durch optionale Studienleistungen erreichten Punkte als Bonuspunkte angerechnet. Eine 1,0 ist in der Klausur auf jeden Fall auch ohne Punkte aus der Bearbeitung optionaler Studienleistungen erreichbar.

<b>Modul ISM-PT-B Methoden der Präsentation, Gesprächsführung und Diskussion</b> <i>Methods of presentation, conversation and discussion</i>		3 ECTS / 90 h
(seit WS19/20) Modulverantwortliche/r: Dr. Ulrich Jentzsch		
<b>Inhalte:</b> An Beispielen von Präsentationen, Einzelgesprächen und Diskussionen sollen <ul style="list-style-type: none"> <li>• persönliche Wirkung auf einzelne und Gruppen</li> <li>• formale und gruppensdynamische Abläufe und</li> <li>• inhaltliche Darstellungsformen</li> </ul> bewusst gemacht und zielbezogen für Präsentationen, für Gespräche und für Diskussionen geübt werden.		
<b>Lernziele/Kompetenzen:</b> Die persönliche Wirkung auf Einzelpersonen und Gruppen kennen lernen und verbessern; Inhalte sachlich verständlich, didaktisch ansprechend und adressatengerecht präsentieren; Kurzvorträge, Gespräche und Diskussionen führen und trainieren.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> keine		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>Methoden der Präsentation, Gesprächsführung und Diskussion</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Dozenten:</b> Dr. Ulrich Jentzsch <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, jährlich		<b>2,00 SWS</b>
<b>Inhalte:</b> Das Seminar ist als Training konzipiert. Methodisch kommen Einzel- und Gruppenübungen sowie Gruppenarbeiten zur Anwendung. Die persönlichen Verhaltensaspekte werden durch Videoaufzeichnungen dokumentiert und anschließend kommentiert.		
<b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten		

<b>Modul ISM-TA-B Technikfolgenabschätzung/-bewertung</b> <i>Technology Assessment</i>		3 ECTS / 90 h
(seit WS19/20) Modulverantwortliche/r: Dr. Ulrich Jentzsch		
<b>Inhalte:</b> Naturwissenschaften, Technik und Wirtschaft haben derzeit wohl den größten Einfluss auf das Denken, das Handeln und die Lebensbedingungen der Menschen in den Industrie- und Schwellenländern. Dieser Einfluss wirkt auf allen Ebenen der Gesellschaft bis auf das unternehmerische Verhalten mittelständischer Firmen. Daher wird anhand eines methodischen Rahmens versucht, aus der Sicht derer, die Naturwissenschaft, Technik und Wirtschaft für sich nutzbringend vorantreiben und aus der Sicht jener, die ohne Nutzen nur Betroffene sind, die wesentlichen Ziele, Kriterien und möglichen Folgen der daraus entstehenden Prozesse zu ermitteln, zu hinterfragen und nach festzulegenden Kriterien zu bewerten. Für die Wahl der zu bearbeitenden Themenfelder ist vor allem auf Grund der Übungen vorab festzulegen, ob der Schwerpunkt bei übergeordneten oder bei betrieblichen Themen liegen soll. Den Ausgangspunkt für die Darstellungen bilden immer aktuelle Themenfelder.		
<b>Lernziele/Kompetenzen:</b> Das besondere Augenmerk liegt dabei auf der untrennbaren Verflechtung von Naturwissenschaft, Technik und Wirtschaft mit ihren Entwicklungsimpulsen einerseits und der Bedeutung der zum Teil konträren weltanschaulichen Überzeugungen von Bevölkerungsgruppen andererseits. Dieses Spannungsverhältnis unterliegt der Technikfolgenbewertung vor allem: <ul style="list-style-type: none"> <li>• durch das internationale Engagement der Unternehmen</li> <li>• den immer weniger widerspruchsfrei akzeptierten Folgen der technisch-wirtschaftlichen Entwicklungen</li> <li>• sowie dem Handicap, komplexe Prozesse mit weltanschaulichen Aspekten nicht durch streng wissenschaftliche Methoden erfassen zu können.</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> keine		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Technikfolgenabschätzung/-bewertung</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Dozenten:</b> Dr. Ulrich Jentzsch <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Vermittlung von Grundkenntnissen: <ul style="list-style-type: none"> <li>• zu den Begriffsinhalten einer TFA/TFB</li> <li>• zu zentralen Themenfeldern und Fragestellungen einer TFA/TFB</li> </ul>	

<ul style="list-style-type: none"><li>• zu den Möglichkeiten und Grenzen prognostischer Aussagen im Rahmen einer TFA/TFB</li><li>• zum prinzipiellen inhaltlichen Aufbau und einer formalen Struktur einer TFA/TFB</li><li>• zu häufig verwendeten Methoden zur Problem-bzw. Entscheidungsanalyse innerhalb einer TFA/TFB-mit Übungen</li></ul>	
---	--

<b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten	
--	--

<b>Modul KInf-GeoInf-B Geoinformationssysteme</b> <i>Geographical Information Systems</i>		6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Christoph Schlieder		
<b>Inhalte:</b> Das Modul führt ein in die Grundlagen der Geoinformationsverarbeitung. Es besteht aus zwei Lehrveranstaltungen: einer Vorlesung, die Konzepte und Methoden vermittelt sowie einer Übung, in der die Anwendung der Methoden auf konkrete Problemstellungen eingeübt wird. Eine weitergehende Inhaltsbeschreibung findet sich bei den Lehrveranstaltungen.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden lernen Grundbegriffe sowie wichtige Methoden aus dem Bereich der Geoinformationssysteme kennen. Sie erwerben folgende Kompetenzen: <ul style="list-style-type: none"> <li>• fachliche Anforderungen im Hinblick auf die Geodatenmodellierung zu analysieren und passende Geodatenmodelle zu erstellen</li> <li>• geoinformatische Analyseverfahren vergleichend zu bewerten und die für ein Anwendungsproblem geeigneten Verfahren zu identifizieren.</li> </ul>		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung: 30 Stunden</li> <li>• Vor- und Nachbereitung der Übung inkl. Recherche und Studium zusätzlicher Quellen aber ohne Bearbeitung der Übungsaufgaben: 30 Stunden</li> <li>• Bearbeiten der Übungsaufgaben: 45 Stunden</li> <li>• Prüfungsvorbereitung: 30 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundkenntnisse der Informatik, wie sie in den empfohlenen Modulen vermittelt werden  Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Informatik und Programmierung für die Kulturwissenschaften (KInf-IPKult-E) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>1. Geoinformationssysteme</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Christoph Schlieder <b>Sprache:</b> Deutsch		<b>2,00 SWS</b>

<p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b>                  Geoinformationssysteme (GIS) dienen der effizienten Erfassung, Analyse und Bereitstellung georeferenzierter Daten. Die Lehrveranstaltung stellt die grundlegenden Konzepte vor, die der Modellierung von Geodaten zugrunde liegen. Hierzu gehört z.B. die unterschiedliche Repräsentation räumlicher Objekte in Vektor- und Raster-GIS. Weitere Themen sind die Geodaten-Erfassung sowie Ansätze zur Geodatenvisualisierung. Anwendungen der Geoinformationsverarbeitung werden an klassischen Einsatzfeldern (Umweltinformationssysteme) und aktuellen technologischen Entwicklungen (mobile Computing) illustriert. Querverbindungen zum Bereich der Semantischen Informationsverarbeitung ergeben sich vor allem im Zusammenhang mit der Interoperabilität von GIS.</p> <hr/> <p><b>Literatur:</b>                  Longley, P., Goodchild, M., Maguire, D., Rhind, D. (2001): Geographic Information: Systems and Science, Wiley: Chichester, UK.                  Shekhar, S., Chawla, S. (2003): Spatial Databases: A Tour, Prentice Hall: Upper Saddle River, NJ.                  Smith, M., Goodchild, M., and Longley, P. (2007): Geospatial Analysis, 2nd edition, Troubador Publishing Ltd.</p>	
<p><b>2. Geoinformationssysteme</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Mitarbeiter Angewandte Informatik in den Kultur-, Geschichts- und Geowissenschaften</p> <p><b>Sprache:</b> Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b>                  siehe Vorlesung</p> <hr/> <p><b>Literatur:</b>                  siehe Vorlesung</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b>                  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b>                  In der schriftlichen Prüfung werden die in Vorlesung und Übung behandelten Themengebiete geprüft.</p>	

<b>Modul KInf-SemInf-M Semantische Informationsverarbeitung</b> <i>Semantic Information Processing</i>		6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(seit WS19/20) Modulverantwortliche/r: Prof. Dr. Christoph Schlieder		
<b>Inhalte:</b> Das Modul führt in das Forschungsgebiet der Semantischen Informationsverarbeitung ein. Es besteht aus zwei Lehrveranstaltungen: einer Vorlesung, die Konzepte und Methoden vermittelt sowie einer Übung, in der die Anwendung der Methoden auf konkrete Problemstellungen eingeübt wird. Eine Inhaltsbeschreibung findet sich bei den Lehrveranstaltungen.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden lernen Grundbegriffe sowie wichtige Methoden aus dem Bereich der Semantischen Informationsverarbeitung kennen. Sie erwerben folgende Kompetenzen: <ul style="list-style-type: none"> <li>• Erklären und Vergleichen grundlegender Konzepte der Semantischen Informationsverarbeitung</li> <li>• Beschreiben und Anwenden von Methoden des Problemlösens und der Wissensrepräsentation</li> <li>• Analyse verschiedener Algorithmen und Bewertung ihrer Eignung für ein Anwendungsszenario</li> <li>• Modellierung insbesondere in kulturwissenschaftlicher Anwendungsszenarien</li> </ul>		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Vorlesung und Übung: 45 Stunden</li> <li>• Vor- und Nachbereitung der Vorlesung: 30 Stunden</li> <li>• Vor- und Nachbereitung der Übung ohne Bearbeitung der Übungsaufgaben: 30 Stunden</li> <li>• Bearbeiten der Übungsaufgaben: 45 Stunden</li> </ul> Prüfungsvorbereitung: 30 Stunden		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Das Modul setzt Programmierkenntnisse sowie Informatik-Grundlagen (Algorithmen, Datenstrukturen, formale Methoden) voraus wie sie im Rahmen eines Bachelorstudiums der Informatik vermittelt werden		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Semantische Informationsverarbeitung</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Christoph Schlieder <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b>	

<p>Semantische Informationsverarbeitung befasst sich mit Problemen, zu deren Lösung Software-Systeme Wissen extrahieren, repräsentieren und verarbeiten müssen. Maschinelles Schlussfolgern erlaubt es, Wissen aus unterschiedlichen Quellen zusammenzuführen. Eine in der Praxis bedeutsame Umsetzung dieses Prinzips sind die Technologien des Semantic Web. Die Vorlesung führt in Methoden und Werkzeuge der semantischen Informationsverarbeitung ein und zeigt, wie sich diese insbesondere in kulturwissenschaftlichen Anwendungsszenarien einsetzen lassen.</p> <p>Methodisch werden drei Schwerpunkte gesetzt. Der erste Teil der Vorlesung behandelt Verfahren des maschinellen Problemlösens. Hier stehen Verfahren der heuristischen Suche im Vordergrund einschließlich spezieller Lösungen z.B. Suchstrategien für Spiele. Der zweite Teil befasst sich mit der Repräsentation von domänenspezifischem Wissen. Themen sind u.a. die Modellierung formaler Ontologien, das Schlussfolgern über Wissensgraphen und die Technologien des Semantic Web. Im dritten Teil stellt die Vorlesung Verfahren der natürlichen Sprachverarbeitung (z.B. Wortmodelle) und der Bildverarbeitung (z.B. Objekterkennung) vor.</p> <p>Die Vorlesung stellt über die Anwendungsszenarien Querbezüge her zur Forschung zu Geoinformationssystemen und Digitalen Bibliotheken.</p>	
<p><b>Literatur:</b>                  Russell, S., Norvig, P. &amp; Davis, E. (2016): Artificial Intelligence. A Modern Approach. 3rd. Upper Saddle River: Prentice Hall.                  Hitzler, P.; Krötzsch, M.; Rudolph, S. (2010): Foundations of Semantic Web technologies. CRC Press</p>	
<p><b>2. Übung Semantische Informationsverarbeitung</b>  <b>Lehrformen:</b> Übung  <b>Dozenten:</b> Mitarbeiter Angewandte Informatik in den Kultur-, Geschichts- und Geowissenschaften  <b>Sprache:</b> Deutsch  <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b>                  Die Übung setzt die in der Vorlesung erworbenen Konzepte und Methoden anhand praktischer Aufgaben um. Die meisten Aufgaben können mit Hilfe von Papier und Stift gelöst werden (Handsimulation), ein kleiner Teil wird mithilfe von Werkzeugen für semantische Modellierung bearbeitet.</p> <hr/> <p><b>Literatur:</b>                  siehe Vorlesung</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b>                  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten  <b>Beschreibung:</b>                  In der schriftlichen Prüfung werden die in Vorlesung und Übung behandelten Themengebiete geprüft.</p>	

<b>Modul KInf-Seminar-B Bachelorseminar Kulturinformatik</b>		3 ECTS / 90 h
<i>Bachelorseminar Kulturinformatik</i>		23 h Präsenzzeit
(seit SS20)		67 h Selbststudium
Modulverantwortliche/r: Prof. Dr. Christoph Schlieder		
<b>Inhalte:</b> Das Modul führt anhand der Forschungsliteratur in grundlegende Methoden aus dem Bereich der Kulturinformatik ein. Die behandelten Problemstellungen stammen aus den Anwendungsfeldern der Angewandten Informatik der Kultur-, Geschichts- und Geowissenschaften.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden erwerben folgende Kompetenzen: <ul style="list-style-type: none"> <li>• eine wissenschaftliche Fragestellung in einem vorher festgelegten Themenbereich aufzustellen</li> <li>• diese Fragestellung selbstständig zu bearbeiten und eigene Lösungskonzepte zu entwickeln</li> <li>• eigene Arbeiten zu präsentieren</li> <li>• eine wissenschaftliche Arbeit zu verfassen</li> </ul>		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Teilnahme an Gruppen- und Einzelbesprechungen: 23 Stunden</li> <li>• Bearbeiten der Praktikumsaufgaben: 57 Stunden</li> <li>• Kolloquiumsvorbereitung: 10 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Allgemeine Informatik-Kenntnisse sowie Interesse an kulturinformatischen Fragestellungen.		<b>Besondere Bestehensvoraussetzungen:</b>
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorseminar Kulturinformatik</b>	<b>2,00 SWS</b>
<b>Lehrformen:</b> Seminar	
<b>Dozenten:</b> Prof. Dr. Christoph Schlieder, Mitarbeiter Angewandte Informatik in den Kultur-, Geschichts- und Geowissenschaften	
<b>Sprache:</b> Deutsch	
<b>Angebotshäufigkeit:</b> SS, jährlich	
<b>Inhalte:</b> Im Rahmen des Bachelor Seminars Kulturinformatik wird ein jeweils von Semester zu Semester wechselndes Themengebiet aus den Kulturinformatik-Modulen Geoinformationssysteme oder Digitale Bibliotheken und Social Computing weiter vertieft. Dies geschieht im Rahmen von Vorträgen und Hausarbeiten zu einer im Vorfeld festgelegten Fragestellung. Dabei steht die selbstständige wissenschaftliche Arbeit im Vordergrund, sowohl schriftlich als auch in der Programmierung.	

---

**Literatur:**

Aktuelle Literatur wird in der Lehrveranstaltung vorgestellt.

**Prüfung**

Hausarbeit mit Referat / Prüfungsdauer: 4 Monate

Bearbeitungsfrist: 20 Minuten

**Beschreibung:**

Das Seminarthema wird in Form einer schriftlichen Seminararbeit (Hausarbeit) bearbeitet sowie im Seminar der Arbeitsprozess und das Arbeitsergebnis vorgestellt (Referat).

<b>Modul KTR-Datkomm-B Datenkommunikation</b> <i>Data communication</i>	6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Udo Krieger	
<p><b>Inhalte:</b></p> <p>Diese Lehrveranstaltung behandelt die technischen Grundlagen der öffentlichen, betrieblichen und privaten Rechnerkommunikation in lokalen Netzen und Weitverkehrsnetzen sowie grundlegende Aspekte ihres Dienstangebots. Es werden die geläufigsten Dienste-, Netz- und Protokollarchitekturen öffentlicher und privater Datenkommunikationsnetze wie das OSI-Referenzmodell bzw. die TCP/IP-Protokollfamilie mit aufgesetzten Dateitransfer, World Wide Web und Multimedia-Diensten vorgestellt.</p> <p>Ferner werden die Grundprinzipien der eingesetzten Übertragungsverfahren, der Übertragungssicherungs- und Steuerungsalgorithmen und der wichtigsten Medienzugriffsverfahren diskutiert, z.B. geläufige Übertragungs- und Multiplexechniken wie FDMA, TDMA, CDMA, Medienzugriffstechniken der CSMA-Protokollfamilie inklusive ihrer Umsetzung in LANs nach IEEE802.x Standards, Sicherungsprotokolle der ARQ-Familie sowie Flusskontrollstrategien mit variablen Fenstertechniken und ihre Realisierung im HDLC-Protokoll.</p> <p>Außerdem werden grundlegende Adressierungs- und Vermittlungsfunktionen in Rechnernetzen wie Paketvermittlung in Routern und Paketverkehrlenkung nach Kürzeste-Wege-Prinzipien bzw. Verkehrlenkung nach dem Prinzip virtueller Wege dargestellt. Darüber hinaus werden die Grundfunktionen der Transportschicht und ihre exemplarische Umsetzung in TCP erläutert.</p>	
<p><b>Lernziele/Kompetenzen:</b></p> <p>Die Studierenden sollen zu eigenständigem Arbeiten im Bereich moderner Kommunikationsnetze befähigt werden. Es werden Grundkenntnisse der Datenkommunikation und die systematische Analyse der verwendeten Algorithmen mit Hilfe eines interaktiven Konzeptes theoretischer und praktischer Übungsaufgaben vermittelt. Die Studierenden lernen, gegebene Implementierungen der vorgestellten Datenkommunikationsverfahren zu analysieren und durch Messungen im Kommunikationslabor ihr Leistungsverhalten zu überprüfen.</p>	
<p><b>Sonstige Informationen:</b></p> <p>Der Arbeitsaufwand gliedert sich grob wie folgt:</p> <ul style="list-style-type: none"> <li>• Teilnahme an Präsenzveranstaltungen (Vorlesung, Übung, Laborbesprechungen): 45 Stunden</li> <li>• Vor- und Nachbereitung von Vorlesungen und Übungen: 100 Stunden</li> <li>• Prüfungsvorbereitung: 35 Stunden</li> </ul>	
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine</p>	
<p><b>Empfohlene Vorkenntnisse:</b></p> <ul style="list-style-type: none"> <li>• erfolgreich abgeschlossene Prüfungen der Grundlagenfächer des Bachelorstudiums, insbesondere Einführung in Algorithmen, Programmierung und Software sowie grundlegende Kenntnisse effizienter Algorithmen</li> <li>• gute Programmierkenntnisse in JAVA (oder C++)</li> </ul> <p>Modul Fortgeschrittene Java Programmierung (DSG-AJP-B) - empfohlen</p>	<p><b>Besondere Bestehensvoraussetzungen:</b> keine</p>

Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (Gdl-Mfl-1) - empfohlen		
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<p><b>1. Datenkommunikation</b>  <b>Lehrformen:</b> Vorlesung  <b>Dozenten:</b> Prof. Dr. Udo Krieger  <b>Sprache:</b> Deutsch/Englisch  <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Inhalte:</b>                  Diese Lehrveranstaltung behandelt die technischen Grundlagen der öffentlichen, betrieblichen und privaten Rechnerkommunikation in lokalen Netzen und Weitverkehrsnetzen sowie grundlegende Aspekte ihres Dienstangebots. Es werden die geläufigsten Dienste-, Netz- und Protokollarchitekturen öffentlicher und privater Datenkommunikationsnetze wie das OSI-Referenzmodell bzw. die TCP/IP-Protokollfamilie mit aufgesetzten Dateitransfer, World Wide Web und Multimedia-Diensten vorgestellt. Ferner werden die Grundprinzipien der eingesetzten Übertragungs-, Übertragungssicherungs- und Steuerungsalgorithmen und des Medienzugriffs diskutiert, z.B. geläufige Übertragungs- und Multiplextechniken wie FDMA, TDMA und CDMA Medienzugriffstechniken der CSMA-Protokollfamilie inklusive ihrer Umsetzung in LANs nach IEEE802.x Standards, Sicherungsprotokolle der ARQ-Familie sowie Flusskontrollstrategien mit variablen Fenstertechniken und ihre Realisierung. Außerdem werden grundlegende Adressierungs- und Vermittlungsfunktionen in Rechnernetzen wie Paketvermittlung in Routern und Paketverkehrslenkung dargestellt. Darüber hinaus werden die Grundfunktionen der Transportschicht und ihre exemplarische Umsetzung in TCP erläutert. Die Bekanntgabe der Lehrsprache erfolgt in der ersten Sitzung der Lehrveranstaltung.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Lean-Garcia, A., Widjaja, I.: Communication Networks, McGraw-Hill, Boston, 2004</li> <li>• Tanenbaum, A. S.: Computernetzwerke, Pearson Studium, München, 4. Aufl., 2003</li> <li>• Kurose, J., Ross, K.W.: Computernetzwerke – ein Top-Down-Ansatz mit Schwerpunkt Internet, Pearson Studium, München, 2014</li> <li>• Comer, D.: Computernetzwerke und Internets, Pearson Studium, München, 2004</li> </ul> <p>Weitere Angaben und Erläuterungen erfolgen in der 1. Vorlesung.</p>	<b>2,00 SWS</b>
<p><b>2. Datenkommunikation</b>  <b>Lehrformen:</b> Übung</p>	<b>2,00 SWS</b>

**Dozenten:** Prof. Dr. Udo Krieger, Mitarbeiter Informatik, insbesondere Kommunikationsdienste, Telekommunikationssysteme und Rechnernetze

**Sprache:** Deutsch/Englisch

**Angebotshäufigkeit:** WS, jährlich

---

**Inhalte:**

Es werden Grundkenntnisse der Datenkommunikation und die systematische Analyse der dabei verwendeten Algorithmen mit Hilfe eines interaktiven Übungskonzeptes aus Haus- und Laboraufgaben vermittelt. Vorlesungsbegleitend werden diese Übungsaufgaben zu folgenden Themen bearbeitet:

- Netzentwurfprinzipien
- OSI-Protokolle
- TCP/IP-Protokollstapel
- Netzelemente
- Datenübertragungssicherungsschicht
- Medienzugriffsschicht

Die Studierenden lernen, gegebene Implementierungen der vorgestellten

Datenkommunikationsverfahren mathematisch und kommunikationstechnisch zu analysieren, durch Messungen ihr Leistungsverhalten zu überprüfen und Vor- bzw. Nachteile der Lösungen zu bewerten.

Die Bekanntgabe der Lehrsprache erfolgt in der ersten Sitzung der

Lehrveranstaltung.

---

**Literatur:**

- Lean-Garcia, A., Widjaja, I.: Communication Networks, McGraw-Hill, Boston, 2004
- Tanenbaum, A. S.: Computernetzwerke, Pearson Studium, München, 4. Aufl., 2003
- Kurose, J., Ross, K.W.: Computernetzwerke – ein Top-Down-Ansatz mit Schwerpunkt Internet, Pearson Studium, München, 2014
- Comer, D.: Computernetzwerke und Internets, Pearson Studium, München, 2004

Weitere Literatur wird in der Übung benannt.

---

**Prüfung**

schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten

**Beschreibung:**

Die Inhalte der Vorlesung sowie die Aufgabenstellungen, Lösungen und Erkenntnisse der Übung, die Haus- und Laboraufgaben beinhaltet, werden in Form einer Klausur geprüft.

Im Verlauf des Semesters können durch die Bearbeitung der Laboraufgaben der Übung und die erfolgreiche Bewertung der entsprechenden Teilleistungen eine maximale Anzahl von Bonuspunkten erworben werden. Diese Bonuspunkte werden bei der Notenvergabe des Moduls berücksichtigt, wobei das Bestehen der Modulprüfung die Voraussetzung für die Berücksichtigung dieser individuell erbrachten Bonuspunkte ist. Die Berechnungs-, Vergabe- und Anrechnungsmodalitäten der Bonuspunkteregelung werden zu Beginn der Lehrveranstaltung festgelegt und den Studierenden zur Kenntnis gebracht. Diese

Bonuspunkte stellen eine freiwillige Zusatzleistung dar. Das Bestehen der Modulprüfung ist auch ohne diese Zusatzleistung möglich. Das Erreichen der Note 1.0 ist ebenfalls ohne die Erbringung dieser Zusatzleistung möglich.

Die Bekanntgabe der Prüfungssprache erfolgt in der ersten Sitzung der Lehrveranstaltung.

Zulässige Hilfsmittel der Prüfung:

- Taschenrechner ohne vollständige alphanumerische Tastatur und Grafikdisplay

<b>Modul KTR-Mfl-2 Mathematik für Informatik 2 (Lineare Algebra)</b>		6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
<i>Mathematics for Computer Science 2 (Linear Algebra)</i>		
(seit WS18/19)		
Modulverantwortliche/r: Prof. Dr. Udo Krieger		
<b>Inhalte:</b> Die Lehrveranstaltung stellt mathematische Grundlagen der Informatik bereit und ist dem Pflichtbereich der Modulgruppe A1 "Mathematische Grundlagen" für Angewandte Informatik/Software Systems Science zugeordnet. Der besondere Bezug zur Angewandten Informatik bzw. Software Systems Science wird in den Vorlesungsbeispielen und Übungen herausgearbeitet.  Es werden grundlegende Methoden und Algorithmen der Gruppen- und Ringtheorie, der linearen Algebra, der Matrizenalgebra, der Lösungstheorie linearer Gleichungssysteme, der Eigenwerttheorie sowie spezifische Anwendungen der Informatik vorgestellt.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden lernen, die grundlegende Methoden und Algorithmen der Lineare Algebra anzuwenden und spezifische Anwendungen der Angewandten Informatik als Probleme der linearen Algebra zu erkennen, zu formulieren und mit Hilfe geeigneter Verfahren zu lösen.		
<b>Sonstige Informationen:</b> Das Modul stellt die Grundlagen für Studierende der Angewandten Informatik und Software Systems Science sowie Studierendende im Nebenfach verwandter Bachelorstudiengänge der Fakultät WIAI bereit.  Der Arbeitsaufwand gliedert sich grob wie folgt: <ul style="list-style-type: none"><li>• Teilnahme an Präsenzveranstaltungen (Vorlesung, Übung, Laborbesprechungen): 45 Stunden</li><li>• Vor- und Nachbereitung von Vorlesungen und Übungen: 100 Stunden</li><li>• Prüfungsvorbereitung: 35 Stunden</li></ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Schulkenntnisse auf dem Niveau eines Mathematik-Vorkurses  Modul Mathematik-Vorkurs-Bachelorstudium (KTR-MVK-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b> 1. - 2.	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Mathematik für Informatik 2 (Lineare Algebra)</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Dozenten:</b> Prof. Dr. Udo Krieger <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>4,00 SWS</b>
<b>Inhalte:</b> Die Lehrveranstaltung stellt mathematische Grundlagen der Informatik bereit und ist dem Pflichtbereich der Modulgruppe A1 "Mathematische Grundlage" für Angewandte Informatik/Software Systems Science zugeordnet. Es werden	

<p>grundlegende Methoden und Algorithmen der Gruppen- und Ringtheorie, der linearen Algebra, der Matrizenalgebra, der Lösungstheorie linearer Gleichungssysteme, der Eigenwerttheorie sowie spezifische Anwendungen der Informatik vorgestellt.</p>	
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• A. Steger: Diskrete Strukturen 1, Springer, Heidelberg, 2002.</li> <li>• G. Golub, C.F. van Loan: Matrix Computations, 3ed., Johns Hopkins, 1996.</li> <li>• D. Hachenberger: Mathematik für Informatiker, Pearson, 2008.</li> <li>• J. Liesen, V. Mehrmann: Lineare Algebra, Springer, Bachelorkurs Mathematik, 2. Auflage, 2015.</li> <li>• B. Pareigis: Linear Algebra für Informatiker, Springer, 2000.</li> <li>• M.P.H. Wolff u.a.: Mathematik für Informatik und Bioinformatik, Springer, 2004.</li> <li>• Weitere Literatur wird in der Vorlesung benannt.</li> </ul>	
<p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b>  Schriftliche Prüfung zu Inhalten der Vorlesung und Übungen im Umfang von 90 Minuten. Zugelassene Hilfsmittel werden in der Lehrveranstaltung bekannt gegeben.</p>	

<b>Modul KTR-SSSProj-B KTR Bachelorprojekt Software Systems Science</b> <i>KTR Bachelor Project Software Systems Science</i>	12 ECTS / 360 h 70 h Präsenzzeit 290 h Selbststudium
(seit WS18/19) Modulverantwortliche/r: Prof. Dr. Udo Krieger	
<b>Inhalte:</b> Wichtige Fertigkeiten bei der Anwendung neuer Kommunikationstechnologien und zur Entwicklung neuer Kommunikationsdienste sind nur durch die Vermittlung praktischer Fähigkeiten und Erfahrungen in teamorientierten Prozessen unter Zeit- und Zielvorgaben industrienahe erlernbar. Die Studierenden werden in der Lehrveranstaltung in einem angeleiteten, aber ansonsten eigenverantwortlich durchgeführten teamorientierten Arbeitsprozess aktuelle Entwicklungsaufgaben aus dem Forschungsbereich der Professur für Informatik bearbeiten.	
<b>Lernziele/Kompetenzen:</b> Die Lehrveranstaltung vermittelt Einblicke in die Entwicklung neuer Dienstarchitekturen und Netztechnologien aus dem Bereich des Internets der nächsten Generation. Ziel ist der Erwerb praktischer Fertigkeiten auf dem Gebiet der IP-gestützten, qualitätsgesicherten Multimediakommunikation und die Fähigkeit, Lösungsvorschläge moderner Dienstarchitekturen im Internet der Zukunft sicher beurteilen zu können. Studierende sollen ein vertieftes Verständnis der bei der Durchführung von Software-Projekten im Bereich Kommunikationsnetze und -dienste auftretenden konzeptionellen und praktischen Probleme wie auch von erfolgsversprechenden Lösungsansätzen dieser Probleme erhalten. Da dies anhand der intensiven Bearbeitung eines Themas aus dem Forschungsbereich der Professur für Informatik in Kleingruppen oder einzeln geschieht, gewinnen die Studierenden wichtige Erfahrungen in der Durchführung kleinerer, forschungsorientierter Projekte von der Grobkonzeption über die Detailplanung bis hin zur Umsetzung und Dokumentation der Ergebnisse in wissenschaftlich ausgerichteten Arbeitsberichten und in der professionellen Präsentation dieser Ergebnisse.	
<b>Sonstige Informationen:</b> Dieses Modul erstreckt sich über 2 Semester und umfasst 2x6=12 ECTS und 2x4=8 SWS. Der Arbeitsaufwand beträgt insgesamt 360 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 30 Std. Recherche, Planung und Teilnahme am Planungsworkshop</li> <li>• 40 Std. Teilnahme an Projekttreffen, einschließlich Tutorien</li> <li>• 180 Std. Durchführung des Projekts (Projektarbeit)</li> <li>• 50 Std. Erstellung des Zwischenberichts (Hausarbeit)</li> <li>• 60 Std. Erstellung des Abschlussberichts, Erstellung und Präsentation der Projektergebnisse (Hausarbeit und Kolloquium)</li> </ul>	
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine	

<b>Empfohlene Vorkenntnisse:</b> <ul style="list-style-type: none"> <li>• gute Kenntnisse in Mathematik für Informatiker 2</li> <li>• mindestens gute JAVA (oder C/C++) Kenntnisse</li> <li>• Kenntnisse der Datenkommunikation im Umfang von KTR-Datkomm-B oder vergleichbare Kenntnisse werden empfohlen</li> <li>• grundlegende methodische Kenntnisse zur Planung und Durchführung von Softwareprojekten, z.B. im Umfang des Moduls "Software EngineeringLab" (SWT-SWL-B), werden empfohlen.</li> </ul> <p>Modul Programmierung komplexer interagierender Systeme (DSG-PKS-B) - empfohlen                  Modul Datenkommunikation (KTR-Datkomm-B) - empfohlen                  Modul Mathematik für Informatik 2 (Lineare Algebra) (KTR-Mfl-2) - empfohlen</p>		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b> 4.	<b>Minimale Dauer des Moduls:</b> 2 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorprojekt Software Systems Science</b> <b>Lehrformen:</b> Bachelorprojekt <b>Dozenten:</b> Prof. Dr. Udo Krieger <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, SS	<b>8,00 SWS</b>
<b>Inhalte:</b> Die Lehrveranstaltung vermittelt Einblicke in die Entwicklung neuer Dienstarchitekturen und Netztechnologien aus dem Bereich des Internets der nächsten Generation. Im Mittelpunkt steht die eigenständige, teamorientierte praktische Umsetzung eines Entwicklungsauftrages unter Verwendung des erworbenen Wissens einzelner Lehrveranstaltungen des Fachgebiets der Professur für Informatik. Die Betriebssystem-Grundausstattung und erforderliche Software-Werkzeuge wie Vyatta-Router, Wireshark, Atheris und RapidStream werden bereitgestellt. Grundlagen der Handhabung werden von den Studierenden im Projekt selbst erarbeitet. Die Lehrveranstaltung erstreckt sich über 2 konsekutive Semester. Die Organisation der Arbeiten erfolgt in einem industrienahen Projektrahmen aus Definitions-, Vorbereitungs-, Implementierungs- und Präsentationsphasen. Dabei soll, wie in realen Projekten üblich, eine inkrementelle Vorgehensweise durchgeführt werden, d.h: <ul style="list-style-type: none"> <li>• Unterteilung der Arbeiten in Arbeitspakete (laboratories/work packages),</li> <li>• ihre Untergliederung in Aufgaben (tasks) und Teilaufgaben (subtasks) mit Meilensteinen</li> <li>• und der Darlegung von Zwischenergebnissen in einem Zwischenbericht nach dem 1. Semester sowie</li> <li>• einem Abschlussbericht mit Abschlusspräsentation der Arbeitsergebnisse in einem Kolloquium im 2. Semester.</li> </ul>	

Es werden Entwicklungsaufgaben zu aktuellen Forschungsfragen im "Future Generation Internet" bearbeitet. Details werden auf der Webseite der Lehrveranstaltung angekündigt. Eine aktuelle Liste der bearbeiteten Themen der Lehrveranstaltung wird in der 1. Besprechung bereitgestellt. Die Bekanntgabe der Lehrsprache erfolgt in der ersten Sitzung der Lehrveranstaltung.

**Prüfung**

schriftliche Hausarbeit / Bearbeitungsfrist: 4 Monate

**Zulassungsvoraussetzung zur Modulprüfung:**

Regelmäßige Teilnahme an der Lehrveranstaltung

**Beschreibung:**

Die Lehrveranstaltung erstreckt sich über 2 konsekutive Semester. Nach dem ersten Semester werden die Leistungen der als Gruppen- oder Einzelarbeit ausgeführten individuellen schriftlichen Ausarbeitung der Projektaufgaben des 1. Semesters bewertet.

Die Bearbeitungsdauer der Hausarbeit des ersten Semesters beträgt 6 Monate.

Die Bekanntgabe der Prüfungssprache sowie der Gewichtung der Prüfungsleistungen erfolgt in der ersten Sitzung der Lehrveranstaltung.

**Prüfung**

Hausarbeit mit Kolloquium / Prüfungsdauer: 30 Minuten

Bearbeitungsfrist: 4 Monate

**Zulassungsvoraussetzung zur Modulprüfung:**

Regelmäßige Teilnahme an der Lehrveranstaltung

**Beschreibung:**

Die Lehrveranstaltung erstreckt sich über 2 konsekutive Semester. Nach dem zweiten Semester werden die Leistungen der als Gruppen- oder Einzelarbeit ausgeführten individuellen schriftlichen Ausarbeitung der Projektaufgaben des 2. Semesters und ihrer Präsentation sowie die Ergebnisse einer abschließenden, individuellen Kolloquiumssprüfung bewertet.

Die Dauer des Kolloquiums beträgt 30 Minuten. Die Bearbeitungsdauer der 2. Hausarbeit beträgt 6 Monate.

Die Gewichtung der Prüfungsleistungen und die Prüfungssprache werden zu Beginn des Semesters bekanntgegeben.

<b>Modul KTR-Sem-B Bachelorseminar zu Kommunikationssystemen und Rechnernetzen</b> <i>Bachelor Seminar Communication Systems and Computer Networks</i>		3 ECTS / 90 h 20 h Präsenzzeit 70 h Selbststudium
(seit SS20) Modulverantwortliche/r: Prof. Dr. Udo Krieger		
<b>Inhalte:</b> Die Studierenden lernen, aktuelle technische Fragestellungen aus den Themenbereichen der Kommunikationsnetze und -dienste der neuesten Generation anhand der Fachliteratur unter Anleitung wissenschaftlich zu bearbeiten und das erworbene Wissen in systematischer Form schriftlich und mündlich darzulegen.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden lernen, aktuelle technische Fragestellungen aus dem Themenfeld der Kommunikationsnetze und -dienste anhand der Fachliteratur unter Anleitung wissenschaftlich zu bearbeiten und das erworbene Wissen in systematischer Form schriftlich und mündlich darzulegen. Die Fähigkeit zur kritischen Bewertung komplexer technischer Inhalte nach wissenschaftlichen Grundsätzen der Informatik stellt ein wichtiges Lernziel dar.		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Präsenzveranstaltungen inkl. Themenvergabe und Besprechungen mit dem Betreuer: 20 Stunden</li> <li>• Bearbeitung des Fachthemas und schriftliche Darstellung: 54 Stunden</li> <li>• Erarbeitung der Präsentation: 16 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine Modul Datenkommunikation (KTR-Datkomm-B) - pflicht		
<b>Empfohlene Vorkenntnisse:</b> Module gemäß der Spezifikationen des Pflichtbereichs sowie solide Kenntnisse der Datenkommunikation		<b>Besondere Bestehensvoraussetzungen:</b>
<b>Angebotshäufigkeit:</b> jährlich nach Bedarf WS und SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorseminar KTR-Bachelor</b> <b>Lehrformen:</b> Seminar <b>Dozenten:</b> Prof. Dr. Udo Krieger <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> jährlich nach Bedarf WS und SS	<b>2,00 SWS</b>
<b>Lernziele:</b> Die Studierenden lernen, aktuelle Fragestellungen aus dem Themenfeld der Kommunikationsnetze und -dienste anhand der Fachliteratur unter Anleitung wissenschaftlich zu bearbeiten und das erworbene Wissen in systematischer Form schriftlich und mündlich darzulegen.	
<b>Inhalte:</b>	

Es werden aktuelle Fragestellungen aus dem Bereich der Kommunikationstechnik und Rechnernetze unter Anleitung bearbeitet . Die aktuelle Themenliste wird auf der Webseite bereitgestellt.

Die Bekanntgabe der Lehrsprache erfolgt in der ersten Sitzung der

Lehrveranstaltung.

Die schriftliche Ausarbeitung erfolgt in LATEX, die mündliche Darstellung im Rahmen einer Powerpoint-, LATEX-Beamer oder PDF-Präsentation auf Basis der schriftlichen Ausarbeitung in möglichst freier Rede und logisch korrekter, verständlicher Form.

**Literatur:**

Die aktuelle Literaturliste wird bei der Vorbesprechung bereitgestellt.

**Prüfung**

Hausarbeit mit Referat / Prüfungsdauer: 30 Minuten

Bearbeitungsfrist: 4 Monate

**Zulassungsvoraussetzung zur Modulprüfung:**

Regelmäßige Teilnahme an der Lehrveranstaltung

**Beschreibung:**

Die Gesamtnote ergibt sich zu gleichen Teilen aus der Bewertung der schriftlichen Ausarbeitung (mit Bearbeitungsdauer von maximal 4 Monaten) und des Referats und muss mit mindestens ausreichend bewertet sein.

Die Bekanntgabe der Prüfungssprache erfolgt in der ersten Sitzung der Lehrveranstaltung.

<b>Modul KogSys-Sem-B Bachelorseminar Kognitive Systeme</b>		3 ECTS / 90 h
<i>Bachelor Seminar Cognitive Systems</i>		
(seit SS20) Modulverantwortliche/r: Prof. Dr. Ute Schmid		
<b>Inhalte:</b> Aufbauend auf den in den Vorlesungen und Übungen des Faches Kognitive Systeme erworbenen Kenntnissen und Fertigkeiten wird im Seminar die eigenständige Erarbeitung und Präsentation eines Themengebiets auf der Basis von wissenschaftlicher Literatur eingeübt. Die Seminarthemen sind aus dem Bereich Künstliche Intelligenz, beispielsweise: Erklärbare Künstliche Intelligenz, Planen und Lernen, Intelligente Tutorssysteme, Erwerb relationaler Konzepte, Analoges Schließen.		
<b>Lernziele/Kompetenzen:</b> – Vertiefte Einarbeitung in eine spezielle Fragestellung aus dem Bereich Künstliche Intelligenz anhand wissenschaftlicher Literatur mit Fokus auf einem konkreten Algorithmus oder einer konkreten Methode anhand eines vorgegebenen Textes – Suche nach wissenschaftlicher Literatur und Bewertung von Qualität und Relevanz – Mündliche Präsentation einer wissenschaftlichen Arbeit – Abfassen einer Forschungspapiers nach vorgegebenem Format entlang einer Forschungsfrage in Englisch – Diskussion von wissenschaftlichen Arbeiten im Seminar		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Kenntnisse entsprechend einem der folgenden Module: Modul Einführung in die Künstliche Intelligenz (AI-KI-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorseminar Kognitive Systeme</b> <b>Lehrformen:</b> Seminar <b>Dozenten:</b> Prof. Dr. Ute Schmid <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Erarbeitung eines ausgewählten Themas aus dem Bereich Künstliche Intelligenz. Die Lehrsprache wird in der ersten Lehrveranstaltung bekanntgegeben.	
<b>Literatur:</b> wird zu Beginn des Seminars bekanntgegeben	
<b>Prüfung</b> Hausarbeit mit Referat / Prüfungsdauer: 30 Minuten Bearbeitungsfrist: 4 Monate	

**Beschreibung:**

Schriftliche Ausarbeitung zu dem im Seminar bearbeiteten Thema.

Die Prüfungssprache wird in der ersten Lehrveranstaltung bekanntgegeben.

Der Arbeitsaufwand für dieses Modul gliedert sich grob wie folgt:

22.5 h Präsenz über 15 Wochen

2.5 h persönliche Besprechungstermine mit dem Dozenten

30 h Erarbeitung der Literatur (inkl. Algorithmen, Systeme)

10 h Vorbereitung der Präsentation

25 h Abfassen der schriftlichen Ausarbeitung

<b>Modul MI-Sem-B Bachelorseminar zur Medieninformatik</b> <i>Media Informatics Seminar [Bachelor]</i>		3 ECTS / 90 h
(seit SS20) Modulverantwortliche/r: Prof. Dr. Andreas Henrich		
<b>Inhalte:</b> Aufbauend auf den in den Vorlesungen und Übungen des Faches Medieninformatik erworbenen Kenntnissen und Fertigkeiten wird in diesem Seminar die eigenständige Erarbeitung und Präsentation von Themengebieten auf Basis der Literatur verfolgt.		
<b>Lernziele/Kompetenzen:</b> Im Seminar werden die Fähigkeiten im Bereich der kritischen und systematischen Literaturrecherche und -betrachtung ebenso weiterentwickelt wie die Kompetenzen in der Präsentation von Fachthemen (schriftlich und im Vortrag) sowie deren Diskussion.		
<b>Sonstige Informationen:</b> Die Lehrveranstaltung wird in <b>Deutsch</b> durchgeführt. Zahlreiche Quellen und Dokumentationen sind aber auf <b>Englisch</b> verfasst. Vorträge und Ausarbeitungen können in Deutsch oder Englisch verfasst werden. Teilnehmerinnen und Teilnehmer müssen an Fachdiskussionen auf Deutsch und Englisch teilnehmen können.  Der <b>Arbeitsaufwand</b> für dieses Modul gliedert sich typischerweise in folgende Bereiche: <ul style="list-style-type: none"> <li>• Teilnahme an den Präsenzveranstaltungen (Themenvergabe, Besprechungen, Präsentationen): ca. 20 Stunden</li> <li>• Literaturrecherche ...: ca. 25 Stunden</li> <li>• Vorbereitung der Präsentation: ca. 15 Stunden</li> <li>• Erstellen der schriftlichen Ausarbeitung: ca. 30 Stunden</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Kenntnisse entsprechend den unten angegebenen Modulen. Details werden in jedem Semester in der Vorbesprechung oder der Vorankündigung bekannt gegeben.  Modul Bachelor AI Teil-Modulgruppe Wissenschaftliches Arbeiten (AI-WissArb-B) - empfohlen Modul Einführung in die Medieninformatik (MI-EMI-B) - empfohlen Modul Web-Technologien (MI-WebT-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>Bachelorseminar Medieninformatik</b> <b>Lehrformen:</b> Proseminar <b>Dozenten:</b> Prof. Dr. Andreas Henrich, Mitarbeiter Medieninformatik <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich		<b>2,00 SWS</b>

<p><b>Inhalte:</b> Im Seminar werden wechselnde aktuelle Forschungsthemen zu den Inhalten der Lehrveranstaltungen bearbeitet. Dabei sind im Regelfall Aspekte mehrerer Lehrveranstaltungen relevant.</p>	
<p><b>Literatur:</b> wird jeweils zu Beginn der Veranstaltung bekannt gegeben</p>	
<p><b>Prüfung</b> Hausarbeit mit Referat / Prüfungsdauer: 30 Minuten</p> <p><b>Beschreibung:</b> Hausarbeit und Referat zu dem im Seminar vom Teilnehmer bzw. von der Teilnehmerin bearbeiteten Thema, inkl. Diskussion</p>	

<b>Modul MI-WAIAI-B Einführung in das wissenschaftliche Arbeiten für Informatik und Angewandte Informatik</b> <i>Introduction to Academic Research &amp; Writing for Computer Science and Applied Computer Science</i>		3 ECTS / 90 h 22 h Präsenzzeit 68 h Selbststudium
(seit SS19) Modulverantwortliche/r: Prof. Dr. Andreas Henrich		
<b>Inhalte:</b> Im Modul werden wesentliche Methoden und Grundsätze des wissenschaftlichen Arbeitens in Informatik und Angewandter Informatik behandelt: <ul style="list-style-type: none"> <li>• Wissenschaft, Ethik, Forschung</li> <li>• Wissenschaftliche Arbeiten</li> <li>• Wissenschaftliches Arbeiten</li> <li>• Projektmanagement</li> <li>• Vortragen &amp; Präsentieren</li> </ul>		
<b>Lernziele/Kompetenzen:</b> Die Studierenden erwerben die Fähigkeit, selbstständig den Kenntnisstand in einem wissenschaftlichen Teilgebiet der Informatik oder Angewandten Informatik an Hand von Literaturempfehlungen zu überprüfen, zu erweitern und sich mit dem Stand der Forschung vertraut zu machen.  Sie erwerben Kenntnisse zu fachspezifischen Methoden der Literatursuche und lernen Systeme kennen, die bei der Erstellung wissenschaftlicher Arbeiten unterstützen.  Allgemein werden die Grundlagen zum wissenschaftlichen Arbeiten, zum Verfassen wissenschaftlicher Arbeiten und zum Präsentieren vermittelt und eingeübt.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundlegende Kenntnisse der Informatik und Angewandten Informatik.  Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen  Modul Einführung in Rechner- und Betriebssysteme (PSI-EiRBS-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b> ab dem 3.	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Einführung in das wissenschaftliche Arbeiten für Informatik und Angewandte Informatik</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Dozenten:</b> Prof. Dr. Andreas Henrich, Mitarbeiter Medieninformatik <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Lernziele:</b> siehe Modulbeschreibung	

**Inhalte:**

siehe Modulbeschreibung

**Literatur:**

- Helmut Balzert, Marion Schröder, Christian Schäfer: *Wissenschaftliches Arbeiten*, W3L GmbH, 2. Auflage, Taschenbuch, 2011, ISBN-13: 978-3868340341
- Manuel René Theisen, *Wissenschaftliches Arbeiten: Erfolgreich bei Bachelor- und Masterarbeit*, Vahlen, Auflage: 17 (2017), ISBN-13: 978-3800653829

**Prüfung**

schriftliche Modulprüfung (Klausur) / Prüfungsdauer: 60 Minuten

<b>Modul MI-WebT-B Web-Technologien</b> <i>Web Technologies</i>		6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Andreas Henrich		
<b>Inhalte:</b> Nach einer Betrachtung der Grundlagen werden die verschiedenen Ebenen der Entwicklung von Web-Anwendungen von HTML und CSS über JavaScript und entsprechende Bibliotheken bis hin zur Serverseite und Frameworks oder Content Management Systemen betrachtet. Aspekte der Sicherheit von Web-Anwendungen werden ebenfalls angesprochen.		
<b>Lernziele/Kompetenzen:</b> Studierende sollen methodische, konzeptuelle und praktische Fähigkeiten und Fertigkeiten zur Erstellung von Web-Applikationen erwerben. Besonderes Augenmerk wird dabei auf Web 2.0 Technologien gelegt. Die Studierenden sind nach Abschluss des Moduls in der Lage, Web-Anwendungen selbstständig mit gängigen Frameworks und Techniken zu entwickeln.		
<b>Sonstige Informationen:</b> Die Lehrveranstaltungen werden in <b>Deutsch</b> durchgeführt. Zahlreiche Quellen und Dokumentationen der Systeme sind aber auf <b>Englisch</b> . Der <b>Arbeitsaufwand</b> für dieses Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"> <li>• Vorlesung: 22,5 Stunden (entspricht den 2 SWS Vorlesung)</li> <li>• Vor- und Nachbereitung der Vorlesung (inkl. Recherche und Studium zusätzlicher Quellen): ca. 30 Stunden</li> <li>• Semesterbegleitendes Üben, Bearbeiten alter Klausuraufgaben, ... zum Vorlesungsstoff: ca. 30 Stunden (inkl. 7,5 Stunden [= 1/3] der 2 SWS Übungsbetrieb)</li> <li>• Bearbeiten der 3 Teilleistungen: insgesamt ca. 60 Stunden (inkl. 15 Stunden [= 2/3] der 2 SWS Übungsbetrieb)</li> <li>• Prüfungsvorbereitung und Prüfung: ca. 37,5 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Grundkenntnisse der Informatik und zu Medienformaten, wie Sie z. B. in den unten angegebenen Modulen erworben werden können. Insbesondere sind auch Kenntnisse in einer imperativen oder objektorientierten Programmiersprache erforderlich. Modul Einführung in Algorithmen, Programmierung und Software (DSG-EiAPS-B) - empfohlen Modul Einführung in die Medieninformatik (MI-EMI-B) - empfohlen		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>1. Web-Technologien</b>		<b>2,00 SWS</b>

<p><b>Lehrformen:</b> Vorlesung  <b>Dozenten:</b> Prof. Dr. Andreas Henrich  <b>Sprache:</b> Deutsch  <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b>  Die Veranstaltung betrachtet die Aufgabenfelder, Konzepte und Technologien zur Entwicklung von Web-Anwendungen. Folgende Bereiche bilden dabei die Schwerpunkte der Veranstaltung:</p> <ul style="list-style-type: none"> <li>• Das Web: Einführung, Architektur, Protokoll ...</li> <li>• Sprachen zur Beschreibung von Webseiten: HTML &amp; CSS</li> <li>• Client-Side Scripting: Basics, AJAX, Bibliotheken</li> <li>• Server-Side Scripting: PHP und weiterführende Konzepte</li> <li>• Frameworks</li> <li>• Sicherheit von Web-Anwendungen</li> <li>• CMS, LMS, SEO &amp; Co.</li> </ul> <hr/> <p><b>Literatur:</b>  aktuelle Literatur wird in der Veranstaltung bekannt gegeben</p>	
<p><b>2. Web-Technologien</b></p> <p><b>Lehrformen:</b> Übung  <b>Dozenten:</b> Mitarbeiter Medieninformatik  <b>Sprache:</b> Deutsch  <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b>  praktische Aufgaben zum Stoff der Vorlesung</p> <hr/> <p><b>Literatur:</b>  siehe Vorlesung</p>	<b>2,00 SWS</b>

<p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b>  <b>Gegenstand</b> der Klausur sind alle Inhalte von Vorlesung und Übung (einschließlich der Teilleistungen; siehe unten).</p> <p>In der <b>Klausur</b> können 90 Punkte erzielt werden.</p> <p>Zusätzlich zur Prüfungsdauer wird eine <b>Lesezeit</b> von 15 Minuten gewährt, um die zu bearbeitenden Aufgaben im Rahmen der Wahlmöglichkeiten auswählen zu können.</p> <p>Im Semester werden studienbegleitend 3 <b>Teilleistungen</b> (schriftliche Hausarbeiten) in der Übung ausgegeben und besprochen, deren Abgabe freiwillig ist. Für jede Teilleistung stehen in der Regel 4 Wochen als Bearbeitungszeit zur Verfügung. Die abgegebenen Lösungen zu den Teilleistungen werden bewertet. Pro Teilleistung können maximal 4 Punkte erzielt werden. Ist die Klausur bestanden (in der Regel sind hierzu 50 % der Punkte erforderlich), so werden die bei der Bearbeitung der Teilleistungen erreichten Punkte (maximal 12</p>	
---	--

---

<p>Punkte) als Bonuspunkte angerechnet. Eine 1,0 ist dabei auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar.</p>	
--	--

<b>Modul MOBI-DBS-B Datenbanksysteme</b> <i>Database Systems</i>		6 ECTS / 180 h
(seit WS20/21) Modulverantwortliche/r: Prof. Dr. Daniela Nicklas		
<b>Inhalte:</b> Das Modul vermittelt eine systematische Einführung in das Gebiet der Datenbanksysteme.		
<b>Lernziele/Kompetenzen:</b> Die Studierenden verstehen die Datenverwaltung auf der Basis des Relationenmodells und kennen grundlegende Architekturkonzepte für Datenmanagementsysteme. Sie erlernen methodische Grundlagen der konzeptuellen Datenmodellierung und verstehen dadurch in vertiefter Weise die Modellierung durch das Entity Relationship Model. Die Studierenden erlernen die Grundlagen der Sprache SQL und können mit SQL Datenbankschemata generieren sowie zugehörige Datenbanken aufbauen und manipulieren. Sie verstehen die Grundlagen von Transaktionssystemen. Schließlich sammeln sie erste Erfahrungen im Umgang mit realen Datenbankverwaltungssystemen.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> keine		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Datenbanksysteme</b> <b>Lehrformen:</b> Vorlesung, Übung <b>Dozenten:</b> Prof. Dr. Daniela Nicklas <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>4,00 SWS</b>
<b>Lernziele:</b> Die Studierenden verstehen die Datenverwaltung auf der Basis des Relationenmodells und kennen grundlegende Architekturkonzepte für Datenmanagementsysteme. Sie erlernen methodische Grundlagen der konzeptuellen Datenmodellierung und verstehen dadurch in vertiefter Weise die Modellierung durch das Entity Relationship Model. Die Studierenden erlernen die Grundlagen der Sprache SQL und können mit SQL Datenbankschemata generieren sowie zugehörige Datenbanken aufbauen und manipulieren. Sie verstehen die Grundlagen von Transaktionssystemen. Schließlich sammeln sie erste Erfahrungen im Umgang mit realen Datenbankverwaltungssystemen.	
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Datenbank-Konzepte und -Architektur</li> <li>• Modellierung von Datenbanken: Das ER- und EER-Modell</li> <li>• Das relationale Modell</li> <li>• Relationale Algebra</li> <li>• SQL (DDL und DML)</li> </ul>	

<ul style="list-style-type: none"> <li>• Normalisierung und Normalformen</li> <li>• Datenbanken im Mehrbenutzerbetrieb: Transaktionssysteme und Recovery</li> <li>• Alternative Entwicklungen im Bereich Datenbanken</li> </ul>	
<p><b>Literatur:</b>  Date C.J.: An Introduction to database systems. 8th Edition, Addison-Wesley, Reading, Massachusetts 2003  Elmasri &amp; Navathe: Grundlagen von Datenbanksystemen, Pearson, 2002</p>	
<p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b>  Zentrale Klausur. Gegenstand der Klausur sind alle Inhalte von Vorlesung und Übung (einschließlich Teilleistungen; siehe unten).</p> <p>Die Prüfung besteht aus 7 Aufgaben, von denen die besten 6 gewertet werden. Zusätzlich zur Prüfungsdauer wird eine Lesezeit von 15 Minuten gewährt, um die zu bearbeitenden Aufgaben im Rahmen der Wahlmöglichkeiten auswählen zu können.</p> <p>Im Semester werden studienbegleitend Teilleistungen ausgegeben und besprochen, deren Abgabe freiwillig ist. Für die Bearbeitung dieser Teilleistungen können Bonuspunkte vergeben werden. Die Anzahl und Bedingungen der zu erreichenden Bonuspunkte sowie deren Umrechnungsfaktor in mögliche Klausurpunkte werden in der ersten Übungsstunde bekannt gegeben.</p> <p>Ist die Klausur bestanden (in der Regel sind hierzu 50% der Punkte erforderlich), so werden die bei der Bearbeitung der Teilleistungen erreichten, ggf. umgerechneten, Punkte zusätzlich angerechnet. Die Note 1,0 ist auch ohne Punkte aus Teilleistungen erreichbar.</p>	

<b>Modul MOBI-MSS-B Mobility in Software Systems</b> <i>Mobility in Software Systems</i>		6 ECTS / 180 h 45 h Präsenzzeit 135 h Selbststudium
(seit WS20/21) Modulverantwortliche/r: Prof. Dr. Daniela Nicklas		
<b>Inhalte:</b> This lecture covers architectures, implementation techniques and algorithms for mobile software systems and software systems that manage mobility. This includes client-side aspects (mobile applications like location-based services), server-side aspects (data management of moving objects), and aspects of distribution (data communication). In addition, since many mobile software systems deal with sensitive information like the location of users, aspects of location privacy are covered.		
<b>Lernziele/Kompetenzen:</b> The students will understand the challenges of mobility in software systems, and will be able to apply techniques and methods to realize such systems.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Comprehension of the relational data model, relational algebra, and SQL language, obtained e.g. from the Module MOBI-DBS-B: Datenbanksysteme; Basic programming skills in Java.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Mobility in Software Systems</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Sprache:</b> Englisch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>4,00 SWS</b>
<b>Lernziele:</b> The students will understand the challenges of mobility in software systems, and will be able to apply techniques and methods to realize such systems.	
<b>Inhalte:</b> This lecture covers architectures, implementation techniques and algorithms for mobile software systems and software systems that manage mobility. This includes client-side aspects (mobile applications like location-based services), server-side aspects (data management of moving objects), and aspects of distribution (data communication). In addition, since many mobile software systems deal with sensitive information like the location of users, aspects of location privacy are covered.	

<b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten <b>Beschreibung:</b>	
--	--

Central written exam. The examination language is English.

The exam questions will be in English. The questions can be answered in English or German. The content that is relevant for the exam consists of the content presented in the lecture and in the practical assignments.

The exam consists of 7 tasks of which only 6 will be graded. We grant 15 minutes of reading time additional to the exam time to choose the tasks.

Participants who submit solutions for practical assignments can achieve bonus points. Details regarding the number of assignments, the number of bonus points per assignment, the conversion factor from bonus points to exam points (e.g., 10:1) and the type of assignments will be announced in the first practical assignment session.

If the points achieved in the exam are sufficient to pass the exam on its own (generally, this is the case when at least 50% of the points have been obtained), the converted bonus points will be added to the points achieved in the exam.

The grade 1.0 can be achieved without the bonus points.

<b>Modul MOBI-PRS-B Bachelor Project Mobile Software Systems (SoSySc)</b>		12 ECTS / 360 h
<i>Bachelor Project Mobile Software Systems (SoSySc)</i>		
(seit WS19/20)		
Modulverantwortliche/r: Prof. Dr. Daniela Nicklas		
<b>Inhalte:</b>		
<p>Applications of in mobile software systems, which are taken from current research activities in mobile, context-aware systems and data stream management, are carried out in part individually and in part in small teams of students, from conception, via theoretical and/or practical realization, to evaluation. In particular, the project concerns the development of sound concepts pertaining to the task to be addressed under the given project constraints. This requires studying the current research literature and relevant approaches on the project's topic.</p> <p>An example of a project task would be the conceptual development, the prototypic implementation, and the case-study-driven evaluation of a small sensor-based, mobile system, which would require knowledge from the modul MOBI-DSC-M Data streams and event processing.</p> <p>The tasks in the project will be tailored to Bachelor level.</p>		
<b>Lernziele/Kompetenzen:</b>		
Students will gain knowledge and practical experience with the problems that arise when carrying out practical research and software development in the field of mobile computing and sensor-based applications. This covers modeling methods, project management, and data management development.		
<b>Sonstige Informationen:</b>		
Dauer der Lehrveranstaltung 2 Semester		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
Programming skills (Java preferred), e.g. from the module "DSG-AJP-B"; Software project management, e.g. from the module "SWT-SWL-B Software Engineering Lab"; Scientific research and writing, e.g. from the module "MI-WAIAI-B Wissenschaftliches Arbeiten" or SSS-SRW-M Scientific Research on Writing for Master´s Students; Relational databases and SQL, e.g. from the module MOBI-DBS-B (Datenbanksysteme)		keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		2 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelor project Mobile Software Systems (SoSySc)</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Prof. Dr. Daniela Nicklas <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> WS, SS <b>Inhalte:</b>	<b>8,00 SWS</b>

<p>Durchführung des Projekts.</p> <p>Begleitend werden Tutorien angeboten, innerhalb derer die Studierenden mit themenrelevanten Methoden, Techniken und Softwarewerkzeugen vertraut gemacht werden. Darüber hinaus finden regelmäßige Treffen zwischen Teilnehmerinnen/Teilnehmern und Projektbetreuern statt, indem der Projektfortschritt kontrolliert und fachlich diskutiert wird.</p> <p>Der Fortschritt bzw. die Ergebnisse des Projekts werden von Studierenden in einem Zwischen- und Abschlussbericht dokumentiert. Zudem wird von jeder Kleingruppe – bzw. bei Individualprojekten von der Teilnehmerin bzw. dem Teilnehmer – ein Poster zum Projekt und dessen Ergebnissen erstellt, welches im Rahmen einer Abschlussveranstaltung präsentiert, erläutert und diskutiert wird.</p>	
---	--

<p><b>Prüfung</b> Hausarbeit mit Kolloquium <b>Zulassungsvoraussetzung zur Modulprüfung:</b> regelmäßige Teilnahme an der Lehrveranstaltung <b>Beschreibung:</b> Als Prüfungsleistung ist eine Hausarbeit sowie ein Kolloquium zu erbringen. Die Bearbeitungsfrist der Hausarbeit und die Prüfungsdauer des Kolloquiums werden zu Beginn einer jeden Lehrveranstaltung von der Projektleiterin bzw. dem Projektleiter bekannt gegeben. Production of a written report on the software project carried out (Assignment/Hausarbeit). Discussion of this project report and of the developed artefacts in the context of the wider project topic (Colloquium/Kolloquium). The term of the project report and of the colloquium will be announced at the beginning of each course by the project leader.</p>	
--	--

<b>Modul MOBI-SEM-B Bachelor-Seminar Mobile Software Systems</b> <i>Bachelor-Seminar Mobile Software Systems</i>		3 ECTS / 90 h
(seit WS19/20) Modulverantwortliche/r: Prof. Dr. Daniela Nicklas		
<b>Inhalte:</b> Guided study and presentation of a topic from Mobile Software Systems / Mobility. The overarching theme of the seminar and example topics will be presented at the beginning of the seminar. The participants will either choose an example topic or define an individual topic within that theme and will contribute to the seminar with presentations and discussions.		
<b>Lernziele/Kompetenzen:</b> Scientific writing and presentation; insight in selected topics; the ability to structure and present a scientific topic.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Scientific research and writing, e.g. from the module "IAIWAI-B Wissenschaftliches Arbeiten" or "SSS-SRW-M Scientific Research on Writing for Master´s Students".		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Mobile Software Systems</b> <b>Lehrformen:</b> Seminar <b>Dozenten:</b> Prof. Dr. Daniela Nicklas <b>Sprache:</b> Englisch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b>	
<b>Prüfung</b> Hausarbeit mit Referat <b>Beschreibung:</b> Als Prüfungsleistung ist eine Hausarbeit sowie ein Referat zu erbringen. Die Bearbeitungsfrist der Hausarbeit und die Prüfungsdauer des Referates werden zu Beginn einer jeden Lehrveranstaltung von der Projektleiterin bzw. dem Projektleiter bekannt gegeben. Production of a written report on the seminar topic (Assignment/Hausarbeit). Discussion of this seminar report in the context of the wider seminar topic (presentation/Referat). The term of the seminar report and of the presentation will be announced at the beginning of each course by the project leader.	

<b>Modul PSI-EDS-B Ethics for the Digital Society</b> <i>Ethics for the Digital Society</i>		3 ECTS / 90 h
(seit SS20) Modulverantwortliche/r: Prof. Dr. Dominik Herrmann		
<b>Inhalte:</b> This module introduces students to fundamental concepts of ethics and their application to techniques that shape the digital society. It discusses the influence of current and upcoming technologies and their implications from an ethical perspective. The lecture is accompanied by a series of case studies, which focus on a concrete problem that is to be analyzed by the participants. Topics include decision making in autonomous systems and systems that employ so-called artificial intelligence, the reliability and dependability of computer systems, and privacy aspects of information systems.		
<b>Lernziele/Kompetenzen:</b> Participants will be able to reflect on their actions as a scientist as well as a computer professional. They learn how to evaluate the trade-offs that are inherent in new technologies and how to design information systems in ways that support the needs of a digital society. Successful participants will obtain the ability to apply ethical thinking to novel problems and potential solutions.		
<b>Sonstige Informationen:</b> The module is taught in English unless all participants are fluent in German. There may be a small number of guest lectures that is taught in German.  During the semester multiple case studies will be published. Participants will be asked to submit essays or solutions (small programs) discussing ethical aspects of those case studies. Essays will be peer-reviewed by other participants.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> keine		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Ethics for the Digital Society</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Dominik Herrmann <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>2,00 SWS</b>
<b>Lernziele:</b> cf. module description	
<b>Inhalte:</b> cf. module description	
<b>Literatur:</b>	
<ul style="list-style-type: none"> <li>Ibo van de Poel and Lamber Royackers: Ethics, Technology, and Engineering – an Introduction</li> </ul>	

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• Jay Quinn: Ethics for the Information Age</li><li>• Herman T. Tavani: Ethics and Technology: Controversies, Questions, and Strategies for Ethical Computing</li></ul> |  |
|---|--|

**Prüfung**

schriftliche Prüfung (Klausur) / Prüfungsdauer: 60 Minuten

**Beschreibung:**

The exam questions will be in English. The questions can be answered in English or German. The content that is relevant for the exam consists of the content presented in the lecture and in the case studies.

The maximum number of points that can be achieved in the exam is 100. Participants that submit all case study essays can collect up to 10 bonus points. Details regarding the number of assignments, the number of points per assignment, and the type of assignments will be announced in the first lecture.

If the points achieved in the exam are sufficient to pass the exam on its own (generally, this is the case when at least 50 points have been obtained), the bonus points will be added to the points achieved in the exam. The grade 1.0 can be achieved without the bonus points.

<b>Modul PSI-EiRBS-B Einführung in Rechner- und Betriebssysteme</b> <i>Introduction to Computer Architecture and Operating Systems</i>	6 ECTS / 180 h
(seit SS21) Modulverantwortliche/r: Prof. Dr. Dominik Herrmann	
<p><b>Inhalte:</b></p> <p>Das Modul bietet einen ersten Einblick in die Informatik der Systeme. Neben einer an Systemen ausgerichteten Einführung in die Informatik behandelt die Veranstaltung die Aufgaben und Architekturmerkmale sowie die wesentlichen Komponenten von Rechner- und Betriebssystemen. Behandelt werden insbesondere der Aufbau und die Funktionsweise eines minimalen Rechners (von-Neumann-Architektur) sowie die Darstellung von Daten (Zahlen und Zeichenketten) im Rechner sowie ihre Speicherung und Verarbeitung. Darüber hinaus werden die wesentlichen Komponenten der Systemsoftware (insbes. Prozess-Scheduling und Speicherverwaltung) erläutert und deren Zusammenspiel mit der Rechnerarchitektur aufgezeigt. Die Themen werden anhand von Modellen, marktgängigen Programmiersprachen (insbes. Java, Python, C) und aktuellen Rechner- und Betriebssystemen (x86 bzw. Linux) behandelt. Abschließend wird ein erster Einblick in Rechnernetze und Aspekte der Systemsicherheit gegeben.</p> <p>Im Rahmen der Veranstaltung erarbeiten sich die Studierenden praktische Kenntnisse im Umgang mit der Linux-Kommandozeile sowie der Assembly-Programmierung. Diese Inhalte erarbeiten sich die Studierenden anhand von bereitgestellten Materialien (Skript und weitere Literatur) und Aufgaben primär im Selbststudium.</p>	
<p><b>Lernziele/Kompetenzen:</b></p> <p>Studierende haben einen ersten Überblick über die Gebiete der Informatik und kennen die grundlegenden Begriffe und Methoden der Informatik sowie die wichtigsten in der Informatik verwendeten Techniken. Die Studierenden haben ein grundlegendes Verständnis zustandsbasierter Systeme und der darin möglichen Abläufe (Prozesse). Zusätzlich kennen sie den Aufbau moderner Rechner- und Betriebssysteme und die dabei zur Anwendung kommenden Informatiktechniken. Die Studierenden sind dazu in der Lage, auf der Linux-Kommandozeile grundlegende Datenverarbeitungsaufgaben zu erledigen sowie einfache Algorithmen mit Assembly-Instruktionen zu kodieren.</p>	
<p><b>Sonstige Informationen:</b></p> <p>Der Arbeitsaufwand von 180 Std. verteilt sich ausgehend von einem 15 Arbeitswochen dauernden Semester in etwa wie folgt:</p> <ul style="list-style-type: none"> <li>• 22.5 Std. Vorlesungsteilnahme</li> <li>• 22.5 Std. Übungsteilnahme</li> <li>• 60 Std. Bearbeiten von wöchentlichen Übungsaufgaben (d.h. ca. 4 Std./Woche)</li> <li>• 30 Std. Vor- und Nachbereitung (Literatur, Recherchen usw.) von Vorlesung und Übung (d.h. ca. 1.5 Std./Woche; ohne Bearbeiten der Übungsaufgaben)</li> <li>• 45 Std. Vorbereitung auf und Zeit für die Abschlussklausur (unter Annahme der o.g. Arbeitsaufwände während des Semesters)</li> </ul> <p>Bei diesem Angaben handelt es sich um Empfehlungen; es besteht weder in Vorlesung noch Übung Anwesenheitspflicht. Der Gesamtaufwand für das Modul ist aber nur einzuhalten, wenn die o.g. Empfehlung in etwa eingehalten wird.</p>	
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine</p>	

<b>Empfohlene Vorkenntnisse:</b> Es sind keine Vorkenntnisse erforderlich. Erste Erfahrungen im Umgang mit der Linux-Kommandozeile sowie einer Programmiersprache (etwa C oder Java) sind hilfreich.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<p><b>1. Einführung in Rechner- und Betriebssysteme</b></p> <p><b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b> vgl. Modulbeschreibung</p> <hr/> <p><b>Literatur:</b> Zum Bereich Rechnerarchitektur und Betriebssysteme gibt es eine ganze Reihe guter einführender Bücher, die aber alle über den in der Vorlesung behandelten Stoff hinausgehen. Deshalb ist die folgende Liste nur als Hinweis auf ergänzende Literatur gedacht. Die Veranstaltung kann auch ohne diese Bücher erfolgreich absolviert werden. Darüber hinaus wird ein ausführliches Skript zur Verfügung gestellt.</p> <ul style="list-style-type: none"> <li>• Tanenbaum, A.S./Austin, T.: Structured Computer Organization. Addison-Wesley, 2012 (6th)</li> <li>• Murdocca, M./Heuring, V.P.: Computer Architecture and Organization. Prentice Hall 2007 (1th)</li> <li>• Tanenbaum, A.S.: Moderne Betriebssysteme. Pearson Studium 2009 (3rd)</li> <li>• Silberschatz, A./Gagne, G./Galvin, P B.: Operating Systems Concepts. John Wiley and Sons, 2012 (9th)</li> </ul>	<b>2,00 SWS</b>
<p><b>2. Einführung in Rechner- und Betriebssysteme (Übung)</b></p> <p><b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b> In der Übung werden die wichtigsten Konzepte der Vorlesung an theoretischen und praktischen Beispielen (z.B. anhand eines Linux-Systems) veranschaulicht und durch die Besprechung von typischen Aufgaben zum jeweiligen Thema vertieft.</p>	<b>2,00 SWS</b>

<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b> Gegenstand der Klausur sind die Inhalte der Vorlesung, des Skripts und der Übungen.</p>	
--	--

<p><b>Modul PSI-IntroSP-B Introduction to Security and Privacy</b></p> <p><i>Introduction to Security and Privacy</i></p>	<p>6 ECTS / 180 h</p>
<p>(seit WS20/21)</p> <p>Modulverantwortliche/r: Prof. Dr. Dominik Herrmann</p>	
<p><b>Inhalte:</b></p> <p>This module introduces students to fundamental concepts in the fields of information security and the protection of privacy. It provides a broad overview over the most relevant topics from a technical perspective. The focus lies on practical issues that have to be considered when professional and personal information systems are built and operated.</p>	
<p><b>Lernziele/Kompetenzen:</b></p> <p>Successful students will know the mathematical background behind basic cryptographic primitives and be able to explain fundamental concepts of information security and privacy, including classical attacks and defenses. They will be able to apply their knowledge when implementing simple attack programs as well as building and operating defensive techniques.</p>	
<p><b>Sonstige Informationen:</b></p> <p>This module is taught in English. It consists of a lecture and tutorials. During the course of the tutorials there will be theoretical and practical assignments (task sheets). Assignments and exam questions can be answered in English or German.</p> <p>Workload breakdown:</p> <ul style="list-style-type: none"> <li>• Lecture: 22.5 hours (2 hours per week)</li> <li>• Tutorials: 22.5 hours (2 hours per week)</li> <li>• Preparation and studying during the semester: 30 hours</li> <li>• Assignments: 67.5 hours</li> <li>• Preparation for the exam (including the exam itself): 37.5 hours</li> </ul>	
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b></p> <p>keine</p>	
<p><b>Empfohlene Vorkenntnisse:</b></p> <p>It is strongly recommended to take this module only after successful completion of PSI-EIRBS-B, which lays the foundation for PSI-IntroSP-B, i.e., prospective PSI-IntroSP-B participants should be familiar with fundamentals of computer architecture (binary representation of strings and numbers in computers, bitwise operators (such as XOR), operation of a CPU, basics of assembly language), operating systems (memory layout and process management), and computer networks (basic IP routing and addressing, TCP/IP connection establishment). Finally, basic familiarity with the Linux command line is recommended.</p> <p>Moreover, basic familiarity with common web technologies (HTTP, HTML, JavaScript) as well as relational database systems and SQL is a recommended prerequisite. Finally, participants should have working knowledge in at least one programming language (e.g., Python, C, or</p>	<p><b>Besondere Bestehensvoraussetzungen:</b></p> <p>keine</p>

Java) so that they can write small tools for automation purposes on demand.		
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<p><b>1. Introduction to Security and Privacy</b></p> <p><b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Englisch <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Lernziele:</b> cf. module description</p> <hr/> <p><b>Inhalte:</b> Selected topics</p> <ul style="list-style-type: none"> <li>• Security Terminology (protection goals, attacker and attack types)</li> <li>• Authentication and Authorization Fundamentals</li> <li>• Software Security in C and Assembly (e.g., buffer overflows, selected defenses)</li> <li>• Cryptography (e.g., historic ciphers, symmetric and asymmetric cryptosystems, Diffie-Hellman key exchange, TLS protocol)</li> <li>• Network Security (spoofing, denial of service, authentication protocols, intrusion detection systems)</li> <li>• Web Security (attacks and defenses related to the OWASP Top 10 including SQL injections and Cross Site Scripting)</li> <li>• Privacy and Techniques for Data Protection (re-identification risks, anonymization networks, k-anonymity, the idea of differential privacy)</li> </ul> <hr/> <p><b>Literatur:</b> Selected books:</p> <ul style="list-style-type: none"> <li>• A. Shostack: Threat Modelling</li> <li>• W. Stallings: Computer Security: Principles and Practice</li> <li>• J. Erickson: Hacking: The Art of Exploitation</li> </ul>	<b>2,00 SWS</b>
<p><b>2. Introduction to Security and Privacy</b></p> <p><b>Lehrformen:</b> Übung <b>Sprache:</b> Englisch <b>Angebotshäufigkeit:</b> WS, jährlich</p>	<b>2,00 SWS</b>

<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b> The content that is relevant for the exam consists of the content presented in the lecture and tutorials. The exam questions are in English. The exam questions can be answered in English or German.</p>	
--	--

<p><b>Modul PSI-SSSProject-B Software Systems Science</b>  <b>Project: Security and Privacy</b>  <i>Software Systems Science Project: Security and Privacy</i></p>	<p>12 ECTS / 360 h</p>
<p>(seit WS18/19)          Modulverantwortliche/r: Prof. Dr. Dominik Herrmann</p>	
<p><b>Inhalte:</b>          This project is specifically offered for Software Systems Science students. The participants of the project familiarize themselves with security and privacy issues that arise during the development of information systems.          Potential tasks during the project include the development of training scenarios for the Insekta platform, designing and/or participating in "build it – break it – fix it" challenges, and contributing to ongoing research activities of members of the Privacy and Security in Information Systems Group. Typically, participants work on their project in small groups. They carry out required research (mostly) on their own, reading about attacks and defenses in textbooks and research papers. Instructors will provide extensive and on-demand support to enable the participants.          Students who are interested in this project may approach a member of the PSI group in order to learn about currently available topics.</p>	
<p><b>Lernziele/Kompetenzen:</b>          Successful students will be able to explain attacks and defenses from textbooks and research papers. They will also be able to carry out selected attacks in practice and implement defenses with a programming language of their choice.</p>	
<p><b>Sonstige Informationen:</b>          This project is taught in English, unless all participants are fluent in German. The workload of this project is equivalent to 360 hours (spread over two semesters).          Workload breakdown:</p> <ul style="list-style-type: none"> <li>• 20 hrs: Getting familiar with the task, obtaining preliminary knowledge</li> <li>• 60 hrs: Reading papers and researching security vulnerabilities</li> <li>• 30 hrs: Preparing the talks (including time for attendance of other talks)</li> <li>• 200 hrs: Implementation</li> <li>• 50 hrs: Writing project report</li> </ul> <p>Note that there are other projects (PSI-ProjectCAD-M, PSI-ProjectPAD, PSI-ProjectSP-M) with different workloads.</p>	
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>          keine</p>	
<p><b>Empfohlene Vorkenntnisse:</b>          Participants should be familiar with basic concepts in information security and privacy, which can be acquired, for instance, by taking the module "Introduction to Security and Privacy" (PSI-IntroSP-B). This includes basic knowledge about the commonly used security terminology, common types of malware and attacks, buffer overflows and related attacks, cryptography, network security, web security, and concepts of privacy.</p>	<p><b>Besondere Bestehensvoraussetzungen:</b>          keine</p>

Moreover, participants should have practical experience with at least one scripting or programming language such as Python or Java. Experience with Linux environments, web technologies, and network protocols is recommended.		
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 2 Semester

<b>Lehrveranstaltungen</b>	
<b>Software Systems Science Project: Security and Privacy</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> WS, SS	<b>8,00 SWS</b>
<b>Lernziele:</b> cf. module description	
<b>Inhalte:</b> cf. module description	
<b>Literatur:</b> Literature will be announced at the beginning of the project.	

<b>Prüfung</b> schriftliche Hausarbeit / Bearbeitungsfrist: 3 Monate <b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regular attendance <b>Beschreibung:</b> The module examination consists of two module examination segments. The respective weights of the two module examination segments will be announced at the beginning of the semester in which the project starts.  The first segment of the module examination consists of a written report (in English) that includes any source code, datasets, and analysis scripts. The maximum number of points that can be achieved in this part of the module examination is 100.  Optionally, participants can submit intermediary results (in English) to collect up to 20 bonus points. If this part of the module examination is passed on its own (generally, this is the case when at least 50 points are obtained), the bonus points will be added to the points achieved in this part of the module examination. The grade 1.0 can be achieved without the bonus points. Details regarding the number of optional submissions during the semester, their type, the points per submission, and the respective deadlines will be announced in the first session of the project.	
---	--

<b>Prüfung</b> Hausarbeit mit Kolloquium / Prüfungsdauer: 30 Minuten Bearbeitungsfrist: 3 Monate <b>Zulassungsvoraussetzung zur Modulprüfung:</b>	
--	--

Regular attendance

**Beschreibung:**

The module examination consists of two module examination segments. The respective weights of the two module examination segments will be announced at the beginning of the semester in which the project starts.

The second segment of the module examination consists of two parts: Firstly, the participants submit a written report (in English) that includes any source code, datasets, and analysis scripts. Secondly, the participants give a talk in which they defend their work (in English; in German if all participants are fluent in German) by presenting related work, their approach, and results. The maximum number of points that can be achieved in the module examination is 100.

Optionally, participants can submit intermediary results (in English) to collect up to 20 bonus points. If this part of the module examination is passed on its own (generally, this is the case when at least 50 points are obtained), the bonus points will be added to the points achieved in this part of the module examination. The grade 1.0 can be achieved without the bonus points. Details regarding the number of optional submissions during the semester, their type, the points per submission, and the respective deadlines will be announced in the first session of the project.

<b>Modul PSI-Sem-B Seminar Security and Privacy Foundations</b>		3 ECTS / 90 h
<i>Seminar Security and Privacy Foundations</i>		
(seit SS20)		
Modulverantwortliche/r: Prof. Dr. Dominik Herrmann		
<b>Inhalte:</b>		
<p>In this seminar, participants learn how to work with scientific literature (books, papers, and articles), how to write in the language used in science, and how to give a talk about a topic covering security and/or privacy foundations. They give a talk summarizing their findings and write them up in a term paper.</p> <p>While participants are expected to perform the actual research on their own, the instructors provide extensive support throughout the seminar. The following questions will be addressed: how to approach a topic, how to find relevant literature, how to read a paper efficiently, how to write a seminar report, and how to give a good talk?</p> <p>The actual topics are subject to change. A list of available topics is made available before the first session via UnivIS or on the website of the Privacy and Security in Information Systems Group.</p>		
<b>Lernziele/Kompetenzen:</b>		
<p>Participants learn to find, read, and summarize scientific texts. They also learn to assess statements and to discuss them critically. Finally, they learn to write scientific texts and to present their results in a talk.</p> <p>Students who participate in an optional peer review procedure will also learn techniques to give useful feedback to others as well as how to accept feedback for one's own work.</p>		
<b>Sonstige Informationen:</b>		
The default language in this seminar is English, unless all participants are fluent in German.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
<p>Participants are highly encouraged to obtain basic knowledge in software engineering as well as foundations of computing, operating systems, and networks before taking this module. Basic knowledge in most of these topics can be obtained by successful completion of PSI-EiRBS-B. Basic knowledge in information security and privacy (which can be obtained by successful completion of PSI-IntroSP-B) is recommended but not strictly required.</p>		keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		1 Semester
<b>Lehrveranstaltungen</b>		
<b>Seminar Security and Privacy Foundations</b>		<b>2,00 SWS</b>
<b>Lehrformen:</b> Seminar		
<b>Dozenten:</b> Prof. Dr. Dominik Herrmann		
<b>Sprache:</b> Englisch/Deutsch		
<b>Angebotshäufigkeit:</b> SS, jährlich		
<b>Lernziele:</b>		
cf. module description		

**Inhalte:**

cf. module description

**Literatur:**

Selected text books:

- Alley: The Craft of Scientific Writing
- Anderson: Security Engineering
- Pfleeger et al.: Security in Computing
- Stallings & Brown: Computer Security: Principles and Practice
- Strunk & White: The Elements of Style

Other relevant literature is presented in the first session.

**Prüfung**

Hausarbeit mit Referat / Prüfungsdauer: 30 Minuten

Bearbeitungsfrist: 3 Monate

**Zulassungsvoraussetzung zur Modulprüfung:**

Continuous attendance in the seminar sessions is mandatory, cf. §9 (10) APO.

**Beschreibung:**

The module examination consists of two parts, a term paper (in English) and a talk (in English; in German if all participants are fluent in German). The maximum number of points that can be achieved in the module examination is 100. Details regarding the number of points that can be achieved in the talk and in the report will be announced in the first session of the project.

Optionally, participants can submit intermediary results (in English) such as surveys, written reviews for the work of other participants, and a draft of the term paper. Participants can thereby earn 20 bonus points. If the module examination is passed on its own (generally, this is the case when at least 50 points are obtained), the bonus points will be added to the points achieved in the module examination. The grade 1.0 can be achieved without the bonus points.

<b>Modul SME-Phy-B Physical Computing</b> <i>Physical Computing</i>		6 ECTS / 180 h
(seit SS20) Modulverantwortliche/r: Prof. Dr. Diedrich Wolter		
<p><b>Inhalte:</b></p> <p>Die in der physikalischen Umwelt eingebetteten Systeme können durch Sensoren ihre Umgebung erfassen und Wissen über ihren jeweiligen Kontext erlangen. Ziel dieses Kurses ist es, einen Überblick über die Möglichkeiten und Herausforderungen von Anwendungen im Bereich Physical Computing zu vermitteln. Physical Computing ist ein neues Gebiet an der Schnittstelle zu intelligenter Interaktion, eingebetteten Systemen und Smart Environments.</p> <p>Dieses Modul setzt zwei Schwerpunkte: Erstens, Kennenlernen von eingebetteten Sensorsystemen sowie Sammeln praktischer Erfahrung mit deren Programmierung und, zweitens, Algorithmen zur Verarbeitung von Sensordaten mit der Zielsetzung, Handlungen und Ereignisse zu erkennen und zu klassifizieren.</p> <p>Im Rahmen des Kurses programmieren Studierende in Kleingruppen ein eingebettetes System mit Sensorik und untersuchen Algorithmen zur Interpretation der Sensordaten:</p> <ul style="list-style-type: none"> <li>• Modellbasierte Interpretation unsicherer Information</li> <li>• Aktions- und Prozesserkennung mit Markov-Modellen</li> <li>• Sensorfusion mit dem Kalmanfilter</li> </ul>		
<p><b>Lernziele/Kompetenzen:</b></p> <ul style="list-style-type: none"> <li>• Einblick in die Programmierung eingebetteter Systeme</li> <li>• Erfahrung in der hardwarenahen Programmierung sammeln</li> <li>• Übersicht über Sensoren erlangen</li> <li>• Eignung von Sensoren zur Erkennung von Kontext und Umweltprozessen beurteilen</li> <li>• Kennenlernen von Algorithmen zur Interpretation von Sensordaten</li> </ul>		
<p><b>Sonstige Informationen:</b></p> <p>The main language of instruction in this course is German. Lectures and tutorials may be delivered in English on demand.</p>		
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b></p> <p>keine</p>		
<p><b>Empfohlene Vorkenntnisse:</b></p> <p>Elementare Grundkenntnisse der Programmierung sind dringend empfohlen (z.B. Modul AI-AuD-B (vormals MI-AuD-B), DSG-JaP-B), Grundkenntnisse in der Programmiersprache C können hilfreich sein.</p>		<p><b>Besondere Bestehensvoraussetzungen:</b></p> <p>keine</p>
<p><b>Angebotshäufigkeit:</b> SS, jährlich</p>	<p><b>Empfohlenes Fachsemester:</b></p>	<p><b>Minimale Dauer des Moduls:</b></p> <p>1 Semester</p>
<p><b>Lehrveranstaltungen</b></p>		
<p><b>1. Physical Computing</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Diedrich Wolter <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich</p>		<p><b>2,00 SWS</b></p>
<p><b>Lernziele:</b></p>		

<p>siehe Modulbeschreibung</p> <hr/> <p><b>Inhalte:</b> siehe Modulbeschreibung</p> <hr/> <p><b>Literatur:</b> wird in der ersten Vorlesung bekanntgegeben</p>	
<p><b>2. Physical Computing</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Prof. Dr. Diedrich Wolter</p> <p><b>Sprache:</b> Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Lernziele:</b> siehe Modulbeschreibung</p> <hr/> <p><b>Inhalte:</b> Praktische Übungen zu den Inhalten der Vorlesung</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p>	

<b>Modul SME-Sem-B Bachelorseminar zu Smart Environments</b>		3 ECTS / 90 h
<i>Bachelor seminar on Smart Environments</i>		
(seit WS20/21)		
Modulverantwortliche/r: Prof. Dr. Diedrich Wolter		
<b>Inhalte:</b>		
In den Seminaren werden jeweils wechselnde Schwerpunkte aus dem Themengebiet Smart Environments behandelt, das in der Informatik insbesondere die Gebiete Künstliche Intelligenz, technische Informatik, und interaktive Systeme berührt. Darüberhinaus wird das wissenschaftliche Arbeiten geübt und ein Einblick in die vielfältigen Problemstellungen und Lösungsansätze im Gebiet Smart Environments vermittelt.		
<b>Lernziele/Kompetenzen:</b>		
Kompetenzerwerb in den Bereichen kritische und systematische Literaturanalyse, Strukturierung komplexer Sachverhalte, bewertender Vergleich konkurrierender Ansätze. Professionelle Präsentation von Fachthemen. Erlernen des Verfassens wissenschaftlicher Arbeiten.		
<b>Sonstige Informationen:</b>		
The main language of instruction in this course is German. Presentations and reports may also be delivered in English.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
keine		keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorseminar zu Smart Environments</b>	<b>2,00 SWS</b>
<b>Lehrformen:</b> Seminar	
<b>Dozenten:</b> Prof. Dr. Diedrich Wolter	
<b>Sprache:</b> Deutsch/Englisch	
<b>Angebotshäufigkeit:</b> WS, jährlich	
<b>Lernziele:</b>	
siehe Modulbeschreibung	
<b>Inhalte:</b>	
Erarbeitung eines ausgewählten Themas aus dem Bereich Smart Environments, Präsentation und Diskussion von Lösungsansätzen	
<b>Literatur:</b>	
wird in der ersten Veranstaltung bekanntgegeben	

<b>Prüfung</b>	
Referat mit schriftl. Hausarbeit / Prüfungsdauer: 30 Minuten	

---

Bearbeitungsfrist: 4 Monate	
-----------------------------	--

<b>Modul SWT-FSA-B Foundations of Software Analysis</b>		6 ECTS / 180 h
<i>Foundations of Software Analysis</i>		
(seit WS19/20)		
Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b>		
This module introduces students to the mathematical and practical foundations of software analysis, which are at the heart of modern techniques for software verification and compiler optimization and key for enhancing software quality.		
<b>Lernziele/Kompetenzen:</b>		
On completion of this module, students will be able to understand key concepts, techniques and algorithms for software analysis and appreciate the workings of modern software analysis tools.		
<b>Sonstige Informationen:</b>		
The main language of instruction is English. The lectures and practicals may be delivered in German if all participating students are fluent in German.		
The total workload of 180 hrs. is split approximately as follows:		
<ul style="list-style-type: none"> <li>• 45 hrs. attending lectures (Vorlesungen)</li> <li>• 45 hrs. attending practicals (Übungen)</li> <li>• 50 hrs. preparing and reviewing the lectures and practicals, including researching literature, studying material from additional sources</li> <li>• 40 hrs. working on the assignment (Hausarbeit) and preparing for the colloquium (Kolloquium)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
Basic knowledge in discrete mathematics and logics, such as acquired in the module "Mathematik für Informatik 1 (Aussagen- und Prädikatenlogik)" (Gdl-Mfl-1).		keine
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Foundations of Software Analysis</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Gerald Lüttgen <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>3,00 SWS</b>
<b>Inhalte:</b>	
Students will be introduced to the foundations of software analysis and their applications to software verification and code analysis and optimization. Particular emphasis will be put on semantics and abstraction, and their underlying mathematical theories based on lattices and order. The following topics will be covered: (i) inductive definitions and proofs; (ii) semantics of programs; (iii) abstraction and abstract interpretation; (iv) elementary fixed point theory; (v) operational and denotational abstract semantics; (vi)	

<p>software verification based on the methods of Floyd and Hoare; (vii) code analysis and optimization based on data flow analysis; (viii) outlook on advanced, modern aspects of software analysis.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Berghammer, R. Semantik von Programmiersprachen, Berlin, Logos Verlag 2001.</li> <li>• Nielson, H. R. and Nielson, F., Semantics with Applications: An Appetizer, Springer, 2007.</li> <li>• Nielson, F., Nielson, H. R. and Hankin, C. Principles of Program Analysis, Springer, 1999.</li> <li>• Loeckx, J. and Sieber, K. The Foundations of Program Verification, 2nd ed. Wiley, 1987.</li> <li>• Davey, B. A. and Priestley, H. A. Introduction to Lattices and Order, 2nd ed. Cambridge University Press, 2002.</li> <li>• Steffen, B., R�uthing, O. and Isberner, M. Grundlagen der h�oheren Informatik: Induktives Vorgehen. Springer, 2013.</li> </ul>	
<p><b>2. Foundations of Software Analysis</b></p> <p><b>Lehrformen:</b> �bung</p> <p><b>Dozenten:</b> Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshufigkeit:</b> WS, jahrllich</p> <hr/> <p><b>Inhalte:</b></p> <p>The practicals (�bungen) will deepen the concepts and techniques taught in the lectures (Vorlesungen), and apply them to the analysis of small examples of software. They will mainly cover pen-and-paper exercises, but will also introduce students to modern software analysis tools. Emphasis will be put on presenting and discussing the solutions to the exercises by and among the students.</p> <hr/> <p><b>Literatur:</b></p> <p>- see the corresponding lectures -</p>	<p><b>3,00 SWS</b></p>
<p><b>Pr�fung</b></p> <p>Hausarbeit mit Kolloquium / Pr�fungsdauer: 20 Minuten</p> <p>Bearbeitungsfrist: 3 Wochen</p> <p><b>Beschreibung:</b></p> <p>Assignment (Hausarbeit) consisting of questions practicing, reviewing and deepening the knowledge transferred in the lectures and practicals (Vorlesungen und �bungen). The assignment is set in English; students may answer in either English or German.</p> <p>Colloquium (Kolloquium) consisting of questions testing the knowledge transferred in the lectures and practicals (Vorlesungen und �bungen), on the basis of the submitted solutions to the assignment (Hausarbeit). The examination language is either English or German and may be chosen by the student at the colloquium.</p>	

<b>Modul SWT-FSE-B Foundations of Software Engineering</b>		6 ECTS / 180 h
<i>Foundations of Software Engineering</i>		
(seit WS19/20)		
Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b>		
This module teaches the foundations of software engineering that are applicable to various kinds of software systems – from information systems to embedded systems. It focusses on technologies, notations and processes for system specification, design, implementation, and verification and validation.		
<b>Lernziele/Kompetenzen:</b>		
Students will receive an introduction to the common problems and paradigms in, and foundations of, software development. They will also gather conceptual and practical knowledge in the analysis, design and testing of software, with an emphasis on technical aspects of specifying, designing, implementing, verifying and validating software.		
<b>Sonstige Informationen:</b>		
The main language of instruction is English. The lectures and practicals may be delivered in German if all participating students are fluent in German.		
The total workload of 180 hrs. is split approximately as follows:		
<ul style="list-style-type: none"> <li>• 45 hrs. attending lectures (Vorlesungen)</li> <li>• 30 hrs. reviewing the lectures, including researching and studying material from additional sources</li> <li>• 45 hrs. attending practicals (Übungen)</li> <li>• 30 hrs. preparing and reviewing the practicals, including researching and studying material from additional sources</li> <li>• 30 hrs. preparing for the written exam (Klausur)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
Basic knowledge in Computer Science, as well as knowledge in programming in Java and in algorithms and data structures.		keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Foundations of Software Engineering</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Gerald Lüttgen <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>3,00 SWS</b>
<b>Inhalte:</b>	
The lectures (Vorlesungen) provide an introduction to the foundations of software engineering, including commonly used technologies, notations and processes for all software engineering phases. In particular, conceptual and technical aspects of software specification, architecture and design, and verification and validation	

<p>are discussed, such as the Unified Modeling Language (UML) and its semantics, model-driven and pattern-based development, and software testing. Students are also introduced to specific aspects of agile software development.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Sommerville, I. Software Engineering, 10th ed. Pearson, 2016.</li> <li>• Robertson, S. and Robertson, J. Mastering the Requirements Process, 3rd ed. Addison-Wesley, 2012.</li> <li>• Cohn, M. User Stories Applied. Addison-Wesley, 2004.</li> <li>• Stevens, P. and Pooley, R. Using UML - Software Engineering with Objects and Components, 2nd. ed. Addison-Wesley, 2006.</li> <li>• Freeman, E., Robson, E., Sierra, K. and Bates, B. Head First Design Patterns. O'Reilly, 2004.</li> <li>• Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Design. Prentice Hall, 1994.</li> </ul> <p>Further literature will be announced in the lectures.</p>	
<p><b>2. Foundations of Software Engineering</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Prof. Dr. Gerald Lüttgen, Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b></p> <p>The practicals (Übungen) exercise and deepen the conceptual knowledge transferred via the lectures (Vorlesungen), and relay practical knowledge in software engineering.</p> <hr/> <p><b>Literatur:</b></p> <p>- see the corresponding lectures -</p>	<p><b>3,00 SWS</b></p>
<p><b>Prüfung</b></p> <p>schriftliche Prüfung (Klausur) / Prüfungsdauer: 120 Minuten</p> <p><b>Beschreibung:</b></p> <p>Written exam (Klausur) consisting of questions that relate to the contents of the lectures (Vorlesungen) and practicals (Übungen) of this module.</p> <p>The written exam is set in English, while answers may be provided in either English or German. The exam is passed if at least 50% of the available points are reached.</p>	

<b>Modul SWT-MBT-B Modellbasiertes Testen</b> <i>Model-based Testing</i>		3 ECTS / 90 h 30 h Präsenzzeit 60 h Selbststudium
(seit WS19/20) Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<p><b>Inhalte:</b></p> <p>Heutige Softwaresysteme zeichnen sich durch einen steigenden Grad an Komplexität und Vernetzung aus. Dies erfordert eine stetig wachsende Anzahl an Testfällen, um eine hohe Testabdeckung für das zu testende System zu gewährleisten. Darüber hinaus werden klassische Vorgehensmodelle bei der Softwareentwicklung zunehmend durch agile Ansätze abgelöst, wodurch die für die Testfallerstellung und -durchführung zur Verfügung stehende Zeit im Vergleich zu klassischen Vorgehensmodellen stark verkürzt wird. Um nun eine hohe Anzahl an Testfällen auch in kurzer Zeit durchführen zu können, ist Testautomatisierung unerlässlich, was allerdings die manuelle Erstellung von Testskripten für das gewünschte Testautomatisierungswerkzeug erfordert. Diese zumeist zeit- und kostenintensive Tätigkeit kann mithilfe des modellbasierten Testens (MBT) automatisiert werden. Im Fokus von MBT steht die Erstellung von sog. Testmodellen, welche die zu testenden Aspekte eines Systems repräsentieren und als Input für die Testfallgenerierung mittels eines MBT-Werkzeugs dienen. Somit lassen sich Testfallbeschreibungen oder ausführbare Testskripte für Testautomatisierungswerkzeuge erzeugen.</p> <p>Ziel des Moduls ist es, einführende theoretische und praktische Kenntnisse des modellbasierten Testens zu vermitteln. Zum Erwerb der praktischen Kenntnisse erstellen Studierende eigenständig kleine Testmodelle und generieren hieraus ausführbare Testfälle mit einem MBT-Werkzeug.</p>		
<p><b>Lernziele/Kompetenzen:</b></p> <p>Nach erfolgreicher Teilnahme an diesem Modul können Studierende insbesondere:</p> <ul style="list-style-type: none"> <li>• MBT in klassischen und agilen/iterativen Vorgehensmodellen anwenden;</li> <li>• Eigenständig Testmodelle erstellen;</li> <li>• Testselektionskriterien zur Testfallgenerierung definieren und anwenden;</li> <li>• Aus Testmodellen Testfallbeschreibungen und ausführbare Testskripte generieren.</li> </ul>		
<p><b>Sonstige Informationen:</b></p> <p>Der Arbeitsaufwand beträgt 90 Std., welche sich grob wie folgt gliedern:</p> <ul style="list-style-type: none"> <li>• 30 Std. Teilnahme an der Vorlesung und Übung</li> <li>• 45 Std. Vor- und Nachbereitung der Vorlesung und Übung</li> <li>• 15 Std. Vorbereitung auf die Klausur</li> </ul>		
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine</p>		
<p><b>Empfohlene Vorkenntnisse:</b> Grundlegende Kenntnisse in Softwareentwicklung und Modellierung mittels UML, wie sie beispielsweise in dem Modul "Foundations of Software Engineering" (SWT-FSE-B) vermittelt werden.</p>		<p><b>Besondere Bestehensvoraussetzungen:</b> Keine</p>
<p><b>Angebotshäufigkeit:</b> WS, jährlich</p>	<p><b>Empfohlenes Fachsemester:</b></p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester</p>
<p><b>Lehrveranstaltungen</b></p>		
<p><b>Modellbasiertes Testen</b></p>		<p><b>2,00 SWS</b></p>

<p><b>Lehrformen:</b> Vorlesung und Übung <b>Dozenten:</b> N.N. <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, jährlich</p> <hr/> <p><b>Lernziele:</b> Siehe Modul-Lernziele/-Kompetenzen</p> <hr/> <p><b>Inhalte:</b> Im theoretischen Teil dieses Moduls werden folgende Kenntnisse vermittelt:</p> <ol style="list-style-type: none"><li>1. Einführung in MBT: Motivation und Konzept</li><li>2. Gängige Arten von Testmodellen</li><li>3. Testauswahlkriterien und Generierung von Testfällen</li><li>4. Übersicht aktueller MBT-Werkzeuge</li><li>5. Einsatz von MBT in klassischen und agilen Entwicklungsprojekten</li></ol> <p>Im praktischen Teil dieses Moduls werden die im theoretischen Teil vermittelten Aspekte mittels eines MBT-Werkzeugs vertieft.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"><li>• Kramer, A. and Legeard, B. Model-Based Testing Essentials – Guide to the ISTQB(R) Certified Model-Based Tester – Foundation Level. John Wiley &amp; Sons, 2016.</li><li>• Roßner, T., Brandes, C., Götz, H. and Winter, M. Basiswissen Modellbasierter Test. dpunkt-Verlag, 2010.</li><li>• Utting, M. and Legeard, B. Practical Model-Based Testing – A Tools Approach. Morgan Kaufmann, 2007.</li></ul> <p>Weitere Literatur wird zu Modulbeginn bekannt gegeben.</p>	
<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten</p> <p><b>Beschreibung:</b> Die Klausur prüft Wissen und Verständnis der in der Vorlesung und Übung vermittelten Lehrinhalte.</p>	

<b>Modul SWT-PCC-M Principles of Compiler Construction</b>		6 ECTS / 180 h
<i>Principles of Compiler Construction</i>		
(seit WS20/21)		
Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b>		
The module teaches the theoretical and practical principles of compiler construction, from lexical analysis and parsing, to semantic analysis, to code generation and optimisation.		
<b>Lernziele/Kompetenzen:</b>		
On completion of this module, students will be familiar with all phases of a modern compiler – from lexical analysis and parsing, to semantic analysis and finally code generation and code optimisation – and will have a deep understanding of the workings of compilers. As a result, students will be able to use compilers more effectively and learn better debugging practices. Students will also be able to start building compilers on their own.		
<b>Sonstige Informationen:</b>		
The main language of instruction is English. The lectures and practicals may be delivered in German if all participating students are fluent in German.		
The total workload of 180 hrs. is split approximately as follows:		
<ul style="list-style-type: none"> <li>• 30 hrs. attending lectures (Vorlesungen)</li> <li>• 30 hrs. reviewing the lectures, including researching and studying material from additional sources</li> <li>• 30 hrs. attending practicals (Übungen)</li> <li>• 30 hrs. preparing and reviewing the practicals, including researching and studying material from additional sources</li> <li>• 60 hrs. working on the assignment (Hausarbeit) and preparing for the colloquium (Kolloquium)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
Basic knowledge in programming languages, in the theoretical foundations of Computer Science (especially in language theory and automata theory) and in algorithms and data structures.		keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		1 Semester

<b>Lehrveranstaltungen</b>	
<b>1. Principles of Compiler Construction</b>	<b>2,00 SWS</b>
<b>Lehrformen:</b> Vorlesung	
<b>Dozenten:</b> Prof. Dr. Gerald Lüttgen	
<b>Sprache:</b> Englisch/Deutsch	
<b>Angebotshäufigkeit:</b> SS, jährlich	
<b>Inhalte:</b>	
Students will be familiarised with a variety of theoretical and practical concepts, techniques and algorithms employed in compiler construction, which reach from language theory, to automata theory, and to data flow analysis. The lectures will	

<p>focus on the following aspects of compiler construction: lexical analysis, parsing, abstract syntax, semantic analysis, code generation and code optimisation.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Louden, K. C. Compiler Construction: Principles and Practice. Course Technology, 1997.</li> <li>• Aho, A. V., Lam, M. S., Sethi, R. and Ullman, J. D. Compilers: Principles, Techniques, and Tools, 2nd ed. Pearson, 2007.</li> <li>• Fischer, C. N., Cytron, R. K. and LeBlanc Jr., R. J. Crafting a Compiler. Pearson, 2010.</li> <li>• Muchnick, S. S. Advanced Compiler Design and Implementation, Morgan Kaufmann, 1997.</li> </ul>	
<p><b>2. Principles of Compiler Construction</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Prof. Dr. Gerald Lüttgen, Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Inhalte:</b></p> <p>Students will practice the theoretical concepts taught in the lectures by applying them to a variety of exercises, so that they can appreciate the diverse range of foundations that make modern programming languages possible. The exercises will largely be pen-and-paper exercises but may also involve some work using computers. Emphasis will be put on presenting and discussing the solutions to the exercises by and among the students, within the timetabled practicals (Übungen). Students can gain further practical experience in compiler construction by attending one of the modules "Masterprojekt Softwaretechnik und Programmiersprachen" (SWT-PR1-M) or "Masters Project in Software Systems Science" (SWT-PR2-M).</p> <hr/> <p><b>Literatur:</b></p> <p>- see the corresponding lectures -</p>	<p><b>2,00 SWS</b></p>
<p><b>Prüfung</b></p> <p>Hausarbeit mit Kolloquium / Prüfungsdauer: 20 Minuten</p> <p>Bearbeitungsfrist: 3 Wochen</p> <p><b>Beschreibung:</b></p> <p>Assignment (Hausarbeit) consisting of questions practicing, reviewing and deepening the knowledge transferred in the lectures (Vorlesungen) and practicals (Übungen). The assignment is set in English language, while answers may be provided in either English or German.</p> <p>Colloquium (Kolloquium) consisting of questions testing the knowledge transferred in the lectures (Vorlesungen) and practicals (Übungen), on the basis of the submitted solutions to the assignment (Hausarbeit). The colloquium can be held electively in English or German language.</p>	

<b>Modul SWT-PR2-B SWT Bachelorprojekt Software Systems Science</b>		12 ECTS / 360 h
<i>SWT Bachelors Project in Software Systems Science</i>		
(seit WS19/20)		
Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b>		
<p>Ein überschaubares Projekt zu aktuellen Themen auf den Gebieten Modellierung, Programmierung, Entwicklung, Analyse und Verifikation von Softwaresystemen wird in zum Teil gemeinsam, zum Teil arbeitsteilig arbeitenden Kleingruppen von Studierenden – und ggf. auch einzeln – von der Konzeption bis zur Umsetzung und deren Dokumentation und Präsentation durchgeführt. Dabei geht es insbesondere um die Entwicklung tragfähiger und mit den vorgegebenen Rahmenbedingungen kompatibler Konzepte zur Lösung der gestellten Aufgabe. Dies erfordert eine umfangreiche Recherche aktueller Literatur und zum Thema vorgeschlagener Ansätze, sowie eine detaillierte Planung. Ein Beispiel eines möglichen Projektthemas ist die Entwicklung einer Mehrbenutzer-Anwendung mit modernen Web-Technologien, die aus einem Front-End und Back-End besteht.</p>		
<b>Lernziele/Kompetenzen:</b>		
<p>Studierende sollen ein vertieftes Verständnis der bei der Durchführung von Software-Projekten auftretenden konzeptionellen und praktischen Probleme wie auch von erfolgsversprechenden Lösungsansätzen zu diesen Problemen erhalten. Da dies anhand der intensiven Bearbeitung eines aktuellen Themas aus dem Bereich der Softwaretechnik und Programmiersprachen in Kleingruppen – oder ggf. auch einzeln – geschieht, gewinnen die Studierenden wichtige Erfahrungen in der Durchführung kleinerer, akademisch orientierter Projekte von der Grobkonzeption über die Detailplanung bis hin zur Umsetzung und Dokumentation der Ergebnisse in wissenschaftlich ausgerichteten Arbeitsberichten und in der Präsentation dieser Ergebnisse.</p>		
<b>Sonstige Informationen:</b>		
<p>Dieses Modul erstreckt sich über 2 Semester: 2x6=12 ECTS, 2x4=8 SWS. Eine regelmäßige Teilnahme an den Projekttreffen ist erforderlich.</p> <p>Der Arbeitsaufwand beträgt insgesamt 360 Std., welche sich grob wie folgt gliedern:</p> <ul style="list-style-type: none"> <li>• 90 Std. Teilnahme an Projekttreffen, einschließlich Tutorien</li> <li>• 180 Std. Durchführung des Projekts (Projektarbeit)</li> <li>• 20 Std. Erstellung des Zwischenberichts (Hausarbeit)</li> <li>• 60 Std. Erstellung des Abschlussberichts, sowie Erstellung und Präsentation des Projektposters (Hausarbeit und Kolloquium)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
<p>Grundlegende methodische und praktische Kenntnisse zur Planung und Durchführung von Softwareprojekten, z. B. erworben in den Modulen "Foundations of Software Engineering" (SWT-FSE-B) und "Software Engineering Lab" (SWT-SWL-B).</p>		keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b>
		2 Semester

<b>Lehrveranstaltungen</b>	
<p><b>SWT Bachelorprojekt Software Systems Science</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Prof. Dr. Gerald Lüttgen, Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshäufigkeit:</b> WS, SS</p> <hr/> <p><b>Lernziele:</b></p> <p>Werden zu Beginn des Projekts bekannt gegeben.</p> <hr/> <p><b>Inhalte:</b></p> <p>Durchführung des Projekts.</p> <p>Begleitend werden nach Bedarf Tutorien angeboten, innerhalb derer die Studierenden mit themenrelevanten Methoden, Techniken und Softwarewerkzeugen vertraut gemacht werden. Darüber hinaus finden regelmäßige Treffen zwischen Teilnehmerinnen/Teilnehmern und Projektbetreuern statt, indem der Projektfortschritt kontrolliert und fachlich diskutiert wird.</p> <p>Der Fortschritt bzw. die Ergebnisse des Projekts werden von Studierenden in einem Zwischen- und Abschlussbericht dokumentiert. Zudem wird von jeder Kleingruppe – bzw. bei Individualprojekten von der Teilnehmerin bzw. dem Teilnehmer – ein Poster zum Projekt und dessen Ergebnissen erstellt.</p> <hr/> <p><b>Literatur:</b></p> <p>Je nach Projektthematik; wird zu Beginn des Projekts bekannt gegeben.</p>	<p><b>8,00 SWS</b></p>

<p><b>Prüfung</b></p> <p>Hausarbeit mit Kolloquium / Prüfungsdauer: 30 Minuten</p> <p>Bearbeitungsfrist: 12 Wochen</p> <p><b>Zulassungsvoraussetzung zur Modulprüfung:</b></p> <p>Regelmäßige Teilnahme an der Lehrveranstaltung.</p> <p><b>Beschreibung:</b></p> <p>Kolloquium: Fachliche Diskussion auf der Grundlage des im Projekt bearbeiteten Themas, auf der zunächst das zum Projekt angefertigte Poster erläutert wird; hier können auch praktische Projektergebnisse (z. B. lauffähige Software) demonstriert werden. Das Kolloquium kann wahlweise in englischer oder deutscher Sprache abgehalten werden.</p> <p>Hausarbeit: Anfertigen eines schriftlichen Abschlussberichts in englischer oder deutscher Sprache nach abgeschlossener Projektarbeit.</p>	
--	--

<p><b>Prüfung</b></p> <p>schriftliche Hausarbeit / Bearbeitungsfrist: 12 Wochen</p> <p><b>Zulassungsvoraussetzung zur Modulprüfung:</b></p> <p>Regelmäßige Teilnahme an der Lehrveranstaltung.</p> <p><b>Beschreibung:</b></p>	
--	--

---

Anfertigen eines schriftlichen Zwischenberichts in englischer oder deutscher Sprache zum Projekt bis spätestens zum Ende des Semesters, in dem das Projekt begonnen wurde.	
--	--

<b>Modul SWT-RSD-B Reactive Systems Design</b> <i>Reactive Systems Design</i>		6 ECTS / 180 h
(seit WS19/20) Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<p><b>Inhalte:</b></p> <p>Reactive systems are digital systems that continuously react to their environment by reading sensor values, computing output values and emitting those values to actuators. Such systems are embedded in many parts of our daily lives and some must even satisfy stringent real-time requirements. Whether it is a home automation system, a driver's assistance system in a modern car, or sophisticated medical equipment at the hospital, we depend on the reliability, correctness, and quality of these systems' software.</p> <p>This module discusses the theoretical concepts and the engineering practice of the model-driven development of reactive systems software. The module's foci are on the synchronous programming paradigm, on automatic code generation from system models, on techniques for verifying and testing reactive systems, and on deploying and integrating reactive software components on a specific operating system and execution platform.</p>		
<p><b>Lernziele/Kompetenzen:</b></p> <p>On completion of this module, students will be able to understand the context and concepts of reactive systems design. They will be able to employ state-of-the-art techniques for the model-driven engineering of reactive software and to apply methods for testing and verifying reactive systems. Moreover, they will have the competence to deploy and integrate reactive software components on a physical model railway platform, taking timing requirements into account.</p>		
<p><b>Sonstige Informationen:</b></p> <p>The main language of instruction is English.</p> <p>The total workload of 180 hrs. is split approximately as follows:</p> <ul style="list-style-type: none"> <li>• 30 hrs. attending lectures (Vorlesungen)</li> <li>• 30 hrs. attending practicals (Übungen)</li> <li>• 60 hrs. preparing and reviewing the lectures and practicals, including researching literature, studying material from additional sources</li> <li>• 60 hrs. working on the assignment (Hausarbeit) and preparing for the colloquium (Kolloquium)</li> </ul>		
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b></p> <p>keine</p>		
<p><b>Empfohlene Vorkenntnisse:</b></p> <p>Basic knowledge in discrete mathematics and programming, e.g., acquired in the modules "<i>Mathematik für Informatik 1 (Aussagen- und Prädikatenlogik)</i>" (GdI-MfI-1) and "<i>Einführung in Algorithmen, Programmierung und Software</i>" (DSG-EiAPS-B). Knowledge gained in program semantics and verification, e.g., in the module "<i>Foundations of Software Analysis</i>" (SWT-FSA-B), is beneficial but not necessary for following the module's content.</p>		<p><b>Besondere Bestehensvoraussetzungen:</b></p> <p>keine</p>
<p><b>Angebotshäufigkeit:</b> SS, jährlich</p>	<p><b>Empfohlenes Fachsemester:</b></p> <p>4.</p>	<p><b>Minimale Dauer des Moduls:</b></p> <p>1 Semester</p>

<b>Lehrveranstaltungen</b>	
<p><b>1. Reactive Systems Design</b></p> <p><b>Lehrformen:</b> Vorlesung</p> <p><b>Dozenten:</b> Prof. Dr. Gerald Lüttgen, Eugene Yip</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Lernziele:</b></p> <p>– see the module's learning outcomes/competences (Lernziele/Kompetenzen) listed above –</p> <hr/> <p><b>Inhalte:</b></p> <p>Students are introduced to modern model-driven techniques, languages and tools for designing and programming reactive systems. The lectures first motivate reactive systems, present their basic design principles, and examine the synchronous programming paradigm. Then, techniques for verifying design properties via model checking, for automatically transforming design models into running code, and for automated testing are studied. The synchronous language and model-based development environment <i>KIELER SCCharts</i> is used for illustrating key semantic and engineering concepts.</p> <p>Several topics on the deployment and integration of reactive software components on a physical execution platform are also addressed: the timing analysis problem, real-time operating systems and scheduling disciplines. In particular, the logical execution time (LET) programming model is discussed as a means to execute reactive software components together in a semantics-preserving manner, and is exemplified by the synchronous programming language ForeC that supports the LET semantics.</p> <hr/> <p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Lee, E. A., and Seshia, S. A. Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 2nd ed. MIT Press, 2017.</li> <li>• Halbwachs, N. Synchronous Programming of Reactive Systems. Springer, 1993.</li> <li>• Harel, D. and Politi, M. Modeling Reactive Systems with Statecharts. McGraw-Hill, 1998.</li> </ul> <p>Further literature will be announced at the beginning of the module.</p>	<b>2,00 SWS</b>
<p><b>2. Reactive Systems Design</b></p> <p><b>Lehrformen:</b> Übung</p> <p><b>Dozenten:</b> Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen</p> <p><b>Sprache:</b> Englisch/Deutsch</p> <p><b>Angebotshäufigkeit:</b> SS, jährlich</p> <hr/> <p><b>Lernziele:</b></p> <p>– see the module's learning outcomes/competences (Lernziele/Kompetenzen) listed above –</p> <hr/> <p><b>Inhalte:</b></p>	<b>2,00 SWS</b>

<p>The practicals (Übungen) deepen the concepts and techniques taught in the lectures (Vorlesungen) and apply them to the development of reactive software. The latter involves a small programming project of a real model railway system with modern development tools such as <i>KIELER SCCharts</i> and a domain-specific language called <i>BahnDSL</i>.</p>	
<p><b>Literatur:</b> – see the corresponding lectures –</p>	
<p><b>Prüfung</b> Hausarbeit mit Kolloquium / Prüfungsdauer: 20 Minuten Bearbeitungsfrist: 3 Wochen</p> <p><b>Beschreibung:</b> The Assignment (Hausarbeit) consists of questions practicing, reviewing and deepening the knowledge transferred in the lectures and practicals (Vorlesungen und Übungen); questions may also involve the practical use of the development tools introduced in the practicals. The assignment is set in English; students may answer in either English or German.</p> <p>The Colloquium (Kolloquium) consists of questions testing the knowledge transferred in the lectures and practicals (Vorlesungen und Übungen), on the basis of the submitted solutions to the assignment (Hausarbeit). The examination language is either English or German and may be chosen by the student at the Colloquium.</p>	

<b>Modul SWT-SEM-B Seminar Software Engineering and Programming Languages (Bachelor)</b>		3 ECTS / 90 h
<i>Seminar in Software Engineering and Programming Languages (Bachelor)</i>		
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b> Current topics in software engineering and programming languages.		
<b>Lernziele/Kompetenzen:</b> Students will compile and acquire current topics in software engineering and programming languages by carrying out and documenting a guided literature survey, and by preparing and delivering a coherent, comprehensible presentation to their peers.		
<b>Sonstige Informationen:</b> The main language of instruction is English. The seminar may be delivered in German if all participating students are fluent in German. Regular participation in the presentations is required.  The total workload of 90 hrs. is split approximately as follows: <ul style="list-style-type: none"> <li>• 5 hrs. introductory tutorials on the seminar's topic and on scientific methods</li> <li>• 20 hrs. consultations and presentations (Referate), including discussions</li> <li>• 25 hrs. literature research and familiarization and evaluation of literature</li> <li>• 40 hrs. working on the assignment (Hausarbeit) and preparation for the presentation (Referat)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Basic knowledge in software engineering and programming languages.		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>Software Engineering and Programming Languages (Bachelor)</b> <b>Lehrformen:</b> Seminar <b>Dozenten:</b> Prof. Dr. Gerald Lüttgen, Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen <b>Sprache:</b> Englisch/Deutsch <b>Angebotshäufigkeit:</b> WS, SS		<b>2,00 SWS</b>
<b>Inhalte:</b> Various current topics in software engineering and programming languages, which complement and/or extend the technical and methodological aspects of the degree programme's modules related to these fields.		
<b>Literatur:</b> Literature will be allocated according to the topics to be discussed.		

---

**Prüfung**

Hausarbeit mit Referat / Prüfungsdauer: 40 Minuten

Bearbeitungsfrist: 8 Wochen

**Zulassungsvoraussetzung zur Modulprüfung:**

Regular participation in the seminar.

**Beschreibung:**

Assignment (Hausarbeit) consisting of a written report on the topic assigned to the student.

Presentation (Referat) on the topic assigned to the student, including a discussion.

<b>Modul SWT-SSP-B Soft Skills in IT-Projekten</b> <i>Soft Skills for IT Projects</i>		3 ECTS / 90 h 30 h Präsenzzeit 60 h Selbststudium
(seit SS14) Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b> Ziel des Moduls ist es, die in der Praxis der IT-Projekte immer wichtiger werdenden Soft Skills wissenschaftlich und methodisch fundiert zu vermitteln. Die Studierenden lernen, dieses Wissen in der Praxis ziel- und lösungsorientiert anwenden zu können.		
<b>Lernziele/Kompetenzen:</b> Nach erfolgreicher Teilnahme an diesem Modul können Studierende insbesondere: <ul style="list-style-type: none"> <li>• Sich die Bedeutung menschlicher Faktoren in großen IT-Projekten bewusst machen;</li> <li>• Erfolgsfaktoren der Teamarbeit kennen und einschätzen;</li> <li>• Eigenkompetenzen und Kompetenzen anderer wahrnehmen, beurteilen und für die Teamorganisation nutzen;</li> <li>• Muster der Gruppendynamik - insbes. Kommunikationsmuster, Konfliktsituationen und Verantwortungsdiffusion - erkennen und managen.</li> </ul>		
<b>Sonstige Informationen:</b> Der Arbeitsaufwand beträgt 90 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 30 Std. Teilnahme an der Vorlesung und Übung</li> <li>• 45 Std. Vor- und Nachbereitung der Vorlesung und Übung</li> <li>• 15 Std. Vorbereitung auf die Klausur</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Keine		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> SS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Soft Skills in IT-Projekten</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Dozenten:</b> Norbert Seifert <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> SS, jährlich	<b>2,00 SWS</b>
<b>Inhalte:</b> Der Inhalt orientiert sich an der in der Praxis großer IT-Projekte erforderlicher Soft Skills: <ol style="list-style-type: none"> <li>1. Vorsprung durch Menschenkenntnis;</li> <li>2. Teamorganisation und -aufstellung;</li> <li>3. Kommunikation und Konfliktmanagement;</li> <li>4. Motivationsfaktoren und Selbstverantwortung;</li> </ol>	

5. Menschliche Spielregeln großer IT-Projekte.	
<b>Literatur:</b> Literatur wird zu Beginn der Lehrveranstaltung angegeben.	
<b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten <b>Beschreibung:</b> Die Klausur prüft Wissen und Verständnis der in der Vorlesung und Übung vermittelten Lehrinhalte.	

<b>Modul SWT-SWL-B Software Engineering Lab</b> <i>Software Engineering Lab</i>		6 ECTS / 180 h
(seit WS19/20) Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
<b>Inhalte:</b> Small teams of students will conduct a software project, starting from a brief problem description. This involves the application of modern software engineering tools, skills in collaboration and team organisation, and knowledge of processes and techniques for producing software artefacts and associated documents.		
<b>Lernziele/Kompetenzen:</b> Students will develop a piece of medium-sized software in small teams, thereby acquiring practical expertise in software engineering and skills in working in a software development team. In addition, this module deepens the students' programming proficiency and their understanding of flexible software engineering processes and of software and process quality, and familiarises them with the deployment and use of modern software engineering tools.		
<b>Sonstige Informationen:</b> The main language of instruction is English. The practicals may be delivered in German if all participating students are fluent in German. A regular attendance of team meetings and active participation is required throughout.  The total workload of 180 hrs. is split approximately as follows: <ul style="list-style-type: none"> <li>• 20 hrs. attending meetings of the student's team with the lecturer (Dozent) on planning, coordination and feedback</li> <li>• 10 hrs. attending the accompanying practicals/tutorials (Übungen/Tutorials) on software tools</li> <li>• 130 hrs. conducting the team project</li> <li>• 20 hrs. working on the assignment (Hausarbeit) and preparing for the colloquium (Kolloquium)</li> </ul>		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> Basic knowledge in Computer Science and Software Engineering, as well as knowledge in Java programming and in programming in the small.		<b>Besondere Bestehensvoraussetzungen:</b>
<b>Angebotshäufigkeit:</b> WS, jährlich	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Software Engineering Lab</b> <b>Lehrformen:</b> Übung <b>Dozenten:</b> Prof. Dr. Gerald Lüttgen, Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, jährlich	<b>4,00 SWS</b>
<b>Inhalte:</b>	

Each team will carry out a software project, regularly meet with their tutor (Dozent) in order to critically reflect on the team's work, and participate in tutorials that introduce the software engineering tools and some software engineering techniques to be used in this project.

**Literatur:**

- Tudose, C., Tahchiev, P., Leme, F., Massol, V. and Gregory, G. JUnit in Action, 3rd ed. Manning Publications, 2020.
- Loeliger, J. and McCullough, M. Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development, 2nd ed. O'Reilly, 2012.
- Vogel, L. Eclipse IDE. Lars Vogel, 2013. ISBN 3943747042.
- Schwaber, K. and Beedle, M. Agile Software Development with Scrum, Pearson, 2001.
- Cohn, M. User Stories Applied. Addison-Wesley, 2004.

See the description of the module "Foundations of Software Engineering (SWT-FSE-B)" for further literature.

**Prüfung**

Hausarbeit mit Kolloquium / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 2 Wochen

**Zulassungsvoraussetzung zur Modulprüfung:**

Regular participation in the associated practicals

**Beschreibung:**

Assignment (Hausarbeit) involving the compilation of a written project report in English or German language by each team, which shall cover the following topics:

- A description of the team's produced artefacts, plus the electronic submission of the artefacts themselves;
- A description, justification and critical reflection of the employed software engineering processes, methods and techniques in general and in each development phase;
- A description of the team's organisation, the distribution of work and the contributions of each team member.

The submission deadline and the details of the required content and format of this report will be announced at the beginning of the semester.

Colloquium (Kolloquium) consisting of a critical discussion of the team's produced software and project report with respect to the taken design decisions and possible alternatives, the quality of the produced artefacts and documentation, the project's status and completeness, the conduct of testing, and the appropriateness of the employed techniques and processes. The colloquium takes place in the presence of the team as a whole, but each question will be addressed to a specific student so that marks can be individualised. The colloquium can be held electively in English or German language.

Because this module involves a team effort, the examination can only be resit in a winter semester.

<b>Modul SoSySc-Thesis-B Bachelorarbeit in Software Systems Science</b>		12 ECTS / 360 h
<i>Bachelor Thesis in Software Systems Science</i>		
(seit SS19)		
Modulverantwortliche/r: Prof. Dr. Gerald Lüttgen		
Weitere Verantwortliche: Professorinnen und Professoren der Informatik		
<b>Inhalte:</b>		
Das Modul Bachelorarbeit hat einen Umfang von 12 ECTS-Punkten und beinhaltet eine schriftliche Prüfung in Form der Bachelorarbeit. Das Thema der Bachelorarbeit ist einem der in Anhang 2 der Prüfungsordnung genannten Fächer zu entnehmen.		
Auf Antrag der Prüfungskandidatin bzw. des Prüfungskandidaten kann vom Prüfungsausschuss auch ein Thema aus einem anderen Fach zugelassen werden. In diesem Fall ist nachzuweisen, dass das gestellte Thema inhaltlich der Software Systems Science entnommen ist.		
<b>Lernziele/Kompetenzen:</b>		
Mit der Bachelorarbeit soll der Nachweis erbracht werden, dass die Prüfungskandidatin bzw. der Prüfungskandidat in der Lage ist, das gestellte Thema selbstständig mit wissenschaftlichen Methoden zu bearbeiten, indem sie erlerntes Fachwissen unter Verwendung wissenschaftlicher Methoden auf eine vorgegebene Forschungsfrage anwenden.		
Die Prüfungskandidatin bzw. der Prüfungskandidat lernt, sich weitgehend selbstständig in eine wissenschaftliche Fragestellung einzuarbeiten. Sie erarbeiten eigeninitiativ eine wissenschaftliche Arbeit und wenden das im Studium erworbene Wissen gezielt und reflektiert an.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
Die Zulassung setzt voraus, dass Module im Umfang von mindestens 120 ECTS-Punkten erfolgreich absolviert wurden.		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
keine		keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b> ab dem 6.	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Prüfung</b>	
schriftliche Hausarbeit / Bearbeitungsfrist: 4 Monate	

<b>Modul Stat-B-01 Methoden der Statistik I</b> <i>Statistics I</i>		6 ECTS / 180 h
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Susanne Rässler		
<p><b>Inhalte:</b> Im Rahmen des Moduls <i>Methoden der Statistik I</i> werden die wichtigsten Grundlagen und Methoden der deskriptiven (beschreibenden) Statistik vermittelt.</p> <p>Dabei werden gegebene Datenmaterialien durch geeignete Aufbereitungs- und Visualisierungstechniken überschaulich dargestellt sowie Maßzahlen zur Charakterisierung von Verteilungen berechnet und sinnvoll interpretiert. Abschließend werden gerichtete lineare Beziehungen anhand des Verfahrens der linearen Regression quantifiziert.</p>		
<p><b>Lernziele/Kompetenzen:</b> Nach der erfolgreichen Absolvierung des Moduls <i>Methoden der Statistik I</i> sind die Studierenden mit den grundlegenden Methoden der deskriptiven Statistik vertraut. Sie sind dazu in der Lage diese Methoden eigenständig anzuwenden, die Voraussetzungen ihrer Anwendung zu prüfen und deren Ergebnisse in geeigneter Weise sinnvoll zu interpretieren.</p>		
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine</p>		
<p><b>Empfohlene Vorkenntnisse:</b> keine</p>		<p><b>Besondere Bestehensvoraussetzungen:</b> keine</p>
<p><b>Angebotshäufigkeit:</b> WS, SS</p>	<p><b>Empfohlenes Fachsemester:</b> ab dem 1.</p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester Semester</p>
<p><b>Lehrveranstaltungen</b></p>		
<p><b>Methoden der Statistik I</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, SS</p>		<p><b>5,00 SWS</b></p>
<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten <b>Zulassungsvoraussetzung zur Modulprüfung:</b> keine</p>		

<b>Modul Stat-B-02 Methoden der Statistik II</b> <i>Statistics II</i>		6 ECTS / 180 h
(seit WS17/18) Modulverantwortliche/r: Prof. Dr. Susanne Rässler		
<p><b>Inhalte:</b></p> <p>Im Rahmen des <i>Moduls Methoden der Statistik II</i> werden die wichtigsten Grundlagen der Wahrscheinlichkeitsrechnung und der induktiven (schließenden) Statistik vermittelt.</p> <p>Dabei werden grundlegenden Begriffe, Regeln und Gesetzmäßigkeiten der Wahrscheinlichkeitsrechnung eingeführt, wobei vor allem Zufallsvorgänge, die sich durch sog. Zufallsvariablen beschreiben lassen, im Vordergrund des Interesses stehen. Viele aus der deskriptiven Statistik bekannte Größen, wie die Verteilungsparameter, können analog für Zufallsvariablen definiert werden. Außerdem werden mit dem Gesetz der großen Zahlen und dem zentralen Grenzwertsatz zwei für die induktive Statistik besonders wichtige Sätze der Wahrscheinlichkeitsrechnung vorgestellt.</p> <p>Des Weiteren stehen Methoden im Vordergrund, nach denen wahrscheinlichkeitstheoretisch fundierte Rückschlüsse von einer Stichprobe auf die betrachtete Grundgesamtheit möglich sind. Aufbauend auf den zuvor behandelten Grundlagen der Wahrscheinlichkeitstheorie werden Verfahren der Punktschätzung und der Intervallschätzung sowie wichtige Hypothesentests behandelt. Im Anschluss daran folgt ein Überblick über einige weitere interessante Teilgebiete der Statistik, wobei speziell die Methode der Regressionsrechnung ausführlicher besprochen wird.</p>		
<p><b>Lernziele/Kompetenzen:</b></p> <p>Nach der erfolgreichen Absolvierung des Moduls <i>Methoden der Statistik II</i> sind die Studierenden mit den grundlegenden Methoden der Wahrscheinlichkeitsrechnung und der induktiven Statistik vertraut. Sie sind dazu in der Lage diese Methoden eigenständig anzuwenden, die Voraussetzungen ihrer Anwendung zu prüfen und deren Ergebnisse in geeigneter Weise sinnvoll zu interpretieren.</p>		
<p><b>Zulassungsvoraussetzung für die Belegung des Moduls:</b></p> <p>keine</p>		
<p><b>Empfohlene Vorkenntnisse:</b></p> <p>Die vorherige Absolvierung des Moduls Stat-B-01 (Methoden der Statistik I).</p>		<p><b>Besondere Bestehensvoraussetzungen:</b></p> <p>keine</p>
<p><b>Angebotshäufigkeit:</b> WS, SS</p>	<p><b>Empfohlenes Fachsemester:</b></p> <p>ab dem 2.</p>	<p><b>Minimale Dauer des Moduls:</b></p> <p>1 Semester</p>
<p><b>Lehrveranstaltungen</b></p>		
<p><b>Methoden der Statistik II</b> <b>Lehrformen:</b> Vorlesung und Übung <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, SS</p>		<p><b>5,00 SWS</b></p>
<p><b>Prüfung</b> schriftliche Prüfung (Klausur) / Prüfungsdauer: 90 Minuten <b>Zulassungsvoraussetzung zur Modulprüfung:</b> keine</p>		

<b>Modul WI-Seminar-B Bachelorseminar aus der Fächergruppen Wirtschaftsinformatik</b> <i>Bachelor Seminar in Information Systems</i>		3 ECTS / 90 h
(seit WS19/20) Modulverantwortliche/r: Prof. Dr. Sven Overhage		
<b>Inhalte:</b> Eigenständige Erarbeitung und Präsentation eines Themas aus dem Fachgebiet der Wirtschaftsinformatik mit wissenschaftlichen Methoden.		
<b>Lernziele/Kompetenzen:</b> Kompetenzerwerb in den Bereichen kritische und systematische Literaturanalyse, Strukturierung komplexer Sachverhalte, bewertender Vergleich konkurrierender Ansätze. Professionelle Präsentation von Fachthemen. Erlernen des Verfassens wissenschaftlicher Arbeiten.		
<b>Sonstige Informationen:</b> Es ist ein Bachelorseminar aus dem Fachgebiet der Wirtschaftsinformatik zu wählen. Die Seminarthemen werden über die jeweiligen Homepages der Lehrstühle bekannt gegeben.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b> keine		
<b>Empfohlene Vorkenntnisse:</b> keine		<b>Besondere Bestehensvoraussetzungen:</b> keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester

<b>Lehrveranstaltungen</b>	
<b>Bachelorseminar</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch/Englisch <b>Angebotshäufigkeit:</b> WS, SS	<b>2,00 SWS</b>
<b>Literatur:</b> Die Literatur wird zu Beginn eines Seminars bekannt gegeben.	

<b>Prüfung</b> Hausarbeit mit Referat <b>Zulassungsvoraussetzung zur Modulprüfung:</b> Regelmäßige Teilnahme an der Lehrveranstaltung <b>Beschreibung:</b> Als Prüfungsleistung ist eine Hausarbeit sowie ein Referat zu erbringen. Die Bearbeitungsfrist der Hausarbeit und die Prüfungsdauer des Referats werden zu Beginn einer jeden Lehrveranstaltung von der Seminarleiterin bzw. dem Seminarleiter bekannt gegeben.	
---	--

<b>Modul WiMa-B-01a Wirtschaftsmathematik I</b> <i>Mathematics for Business and Economics I</i>		3 ECTS / 90 h
(seit WS18/19 bis WS19/20) Modulverantwortliche/r: Prof. Dr. Christian Aßmann		
<b>Inhalte:</b>		
<ol style="list-style-type: none"> <li><b>Grundlagen:</b> Griechisches Alphabet, Mengenlehre, Zahlbereiche, Ungleichungen, Intervalle, Potenzrechnung, Summenzeichen und Produktzeichen, Binomischer Satz.</li> <li><b>Funktionen einer Variablen:</b> Funktionsbegriff, Verknüpfung von Funktionen, Monotone Funktionen, Umkehrfunktionen.</li> <li><b>Elementare Funktionen:</b> Polynome, Rationale Funktionen, Exponential- und Logarithmusfunktionen, Trigonometrische Funktionen.</li> <li><b>Folgen, Reihen, Grenzwerte, Stetigkeit:</b> Endliche Folgen, Arithmetische und Geometrische Folge mit Beispielen im Rahmen der Kapitalverzinsung und Abdiskontierung, Arithmetische und Geometrische Reihe mit Beispielen im Rahmen der Renten- und Tilgungsrechnung, Rechenregeln für Grenzwerte bei Folgen, Rechenregeln für Grenzwerte bei Reihen, Grenzwerte bei Funktionen, Stetigkeitsbegriff.</li> <li><b>Differentialrechnung für Funktionen einer Variablen:</b> Differenzenquotient, Differentialquotient, Ableitungsregeln, Regel von de l'Hôpital.</li> <li><b>Differentialrechnung für Funktionen mehrerer Variablen:</b> Funktionen mehrerer Variablen, Partielle Differentiation, Partielle Ableitungen höherer Ordnung, Ableitung impliziter Funktionen.</li> <li><b>Optimierung:</b> Lokale und globale Extrema, Krümmung von Funktionen einer Variablen, Krümmung von Funktionen mehrerer Variablen, Bedingungen für Extrema von Funktionen einer Variablen, Bedingungen für Extrema von Funktionen mehrerer Variablen, Sattelpunkte von Funktionen einer Variablen, Sattelpunkte von Funktionen mehrerer Variablen, Optimierung unter Nebenbedingungen, Lagrange Verfahren.</li> </ol>		
<b>Lernziele/Kompetenzen:</b>		
Vermittlung von mathematischen Grundkenntnissen aus dem Gebiet der Analysis. Es werden Grundlagen für das Verständnis und die Beherrschung mathematischer Verfahren und Konzepte geschaffen, welche in weiterführenden wirtschaftswissenschaftlichen und wirtschaftsinformatischen Veranstaltungen zum Einsatz kommen.		
<b>Zulassungsvoraussetzung für die Belegung des Moduls:</b>		
keine		
<b>Empfohlene Vorkenntnisse:</b>		<b>Besondere Bestehensvoraussetzungen:</b>
keine		keine
<b>Angebotshäufigkeit:</b> WS, SS	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>Lehrveranstaltungen</b>		
<b>Wirtschaftsmathematik I</b> <b>Lehrformen:</b> Vorlesung <b>Dozenten:</b> Prof. Dr. Christian Aßmann <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> WS, SS		<b>2,00 SWS</b>
<b>Literatur:</b>		

<ul style="list-style-type: none"> <li>• Jensen, U. (1998), Mathematik für Wirtschaftswissenschaftler, Oldenbourg (München).</li> <li>• Jensen, U. (2001), Klausursammlung zur Mathematik für Wirtschaftswissenschaftler, Oldenbourg (München).</li> <li>• Jensen, U. (2010), Wozu Mathe in den Wirtschaftswissenschaften?, Vieweg + Teubner (Wiesbaden).</li> <li>• Merz, M. und Wüthrich, M. (2013), Mathematik für Wirtschaftswissenschaftler, Vahlen (München).</li> <li>• Merz, M. (2013), Übungsbuch zur Mathematik für Wirtschaftswissenschaftler, Vahlen (München).</li> <li>• Opitz O. (1989), Mathematik, Oldenbourg (München, Wien).</li> <li>• Schwarze J. (1981), Mathematik für Wirtschaftswissenschaftler, Band 1-3, Neue Wirtschaftsbriefe, Herne (Berlin).</li> <li>• Sydsaeter K. und Hammond P. (2004), Mathematik für Wirtschaftswissenschaftler, Pearson Studium (München).</li> <li>• Ruhrländer, M. (2016), Brückenkurs Mathematik, Pearson (München).</li> <li>• Böker, F. (2010), Mathematik für Wirtschaftswissenschaftler: Das Übungsbuch, 3. Auflage, Pearson (München).</li> </ul>	
---	--

<p><b>Prüfung</b>  schriftliche Prüfung (Klausur) / Prüfungsdauer: 60 Minuten  <b>Zulassungsvoraussetzung zur Modulprüfung:</b>  keine</p>	
--	--

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
<b>A1 Mathematische Grundlagen</b>			<b>27</b>		
Stat-B-01	Methoden der Statistik I	WS, SS(1)	6	5 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
Stat-B-02	Methoden der Statistik II	WS, SS(1)	6	5 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
WiMa-B-01a	Wirtschaftsmathematik I	WS, SS(1)	3	2 Vorlesung	schriftliche Prüfung (Klausur) 60 Minuten
Gdl-Mfi-1	Mathematik für Informatik 1 (Aussagen- u. Prädikatenlogik)	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
KTR-Mfi-2	Mathematik für Informatik 2 (Lineare Algebra)	SS, jährlich	6	4 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
	<b>A2 Allgemeine Informatik Grundlagen</b>		<b>36</b>		
DSG-EiAPS-B	Einführung in Algorithmen, Programmierung und Software	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
PSI-EiRBS-B	Einführung in Rechner- und Betriebssysteme	SS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
Gdl-GTI-B	Grundlagen der Theoretischen Informatik	SS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
AI-AuD-B	Algorithmen und Datenstrukturen	SS, jährlich(1)	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
SWT-FSA-B	Foundations of Software Analysis	WS, jährlich	6	3 Vorlesung 3 Übung	Hausarbeit mit Kolloquium 3 Wochen 20 Minuten
MOBI-DBS-B	Datenbanksysteme	SS, jährlich	6	4 Vorlesung, Übung	schriftliche Prüfung (Klausur) 90 Minuten

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
<b>A3 Programmierung und Softwaretechnik (Studienbeginn ab WS15/16)</b>			<b>18 - 42</b>		
<b>Pflichtbereich: A3P Pflichtbereich Grundlagen PuSWT</b>			<b>18</b>		
Gdl-IFP-B	Introduction to Functional Programming	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
SWT-FSE-B	Foundations of Software Engineering	SS, jährlich	6	3 Vorlesung 3 Übung	schriftliche Prüfung (Klausur) 120 Minuten
SWT-SWL-B	Software Engineering Lab	WS, jährlich	6	4 Übung	Hausarbeit mit Kolloquium 2 Wochen 45 Minuten
<b>Pflichtbereich: A3W Wahlpflichtbereich Schwerpunkt PuSWE</b>			<b>0 - 24</b>		
DSG-AJP-B	Fortgeschrittene Java Programmierung	SS, jährlich	3	2 kein Typ gewählt, Übung	Hausarbeit mit Kolloquium 3 Monate 10 Minuten
DSG-JaP-B	Java Programmierung	WS, jährlich(2017/2018)	3	2 Vorlesung, Übung	schriftliche Prüfung (Klausur) 90 Minuten
HCI-IS-B	Interaktive Systeme	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
KInf-SemInf-M	Semantische Informationsverarbeitung	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
MI-WebT-B	Web-Technologien	SS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
SME-Phy-B	Physical Computing	SS, jährlich(1)	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
SWT-MBT-B	Modellbasiertes Testen	WS, jährlich(1)	3	2 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
SWT-PCC-M	Principles of Compiler Construction	SS, jährlich	6	2 Vorlesung 2 Übung	Hausarbeit mit Kolloquium 3 Wochen

## Modultabelle

---

SWT-RSD-B	Reactive Systems Design	SS, jährlich(1)	6	2 Vorlesung 2 Übung	20 Minuten Hausarbeit mit Kolloquium 3 Wochen 20 Minuten
-----------	-------------------------	--------------------	---	------------------------	---

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
<b>A4 Komplexe und Verteilte Systeme</b>			<b>21 - 42</b>		
<b>Pflichtbereich: A4P Pflichtbereich Grundlagen KuVS</b>			<b>21</b>		
DSG-PKS-B	Programmierung komplexer interagierender Systeme	WS, jährlich	3	2 Übung	Hausarbeit mit Kolloquium 3 Monate 10 Minuten
KTR-Datkomm-B	Datenkommunikation	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
DSG-IDistrSys-B	Introduction to Distributed Systems	SS, jährlich(2020)	6	2 Vorlesung 2 Übung	Hausarbeit mit Kolloquium 3 Monate 10 Minuten
MOBI-MSS-B	Mobility in Software Systems	WS, jährlich(1)	6	4 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
<b>Wahlpflichtbereich: A4W Wahlpflichtbereich Komplexe und Verteilte Systeme</b>			<b>0 - 21</b>		
AI-KI-B	Einführung in die Künstliche Intelligenz	SS, jährlich(1)	6	2 Vorlesung 2 Übung	schriftliche Modulprüfung (Klausur) 90 Minuten
Gdl-MTL	Modal and Temporal Logic	WS, jährlich	6	4 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
HCI-KS-B	Kooperative Systeme	SS, jährlich	6	2 Vorlesung 2 Übung	mündliche Prüfung 30 Minuten schriftliche Prüfung (Klausur) 90 Minuten
HCI-US-B	Ubiquitäre Systeme	WS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Modulprüfung (Klausur) 90 Minuten mündliche Prüfung 30 Minuten
KInf-GeoInf-B	Geoinformationssysteme	SS, jährlich	6	2 Vorlesung	schriftliche Prüfung (Klausur)

## Modultabelle

---

PSI-IntroSP-B	Introduction to Security and Privacy	WS, jährlich(1)	6	2 Übung 2 Vorlesung 2 Übung	90 Minuten schriftliche Prüfung (Klausur) 90 Minuten
---------------	--------------------------------------	--------------------	---	-----------------------------------	--

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
	<b>A5 Fachstudium Anwendungsfächer</b>		<b>12 - 20</b>		
	<p>Anwendungsfächer mit Bezug zu Software Systems Science sind Wahlpflichtmodule anderer Fakultäten im Umfang von insgesamt 12 bis 20 ECTS-Punkten aus Bachelornebenfächern gemäß des Anhangs der Allgemeinen Prüfungsordnung für Bachelor- und Masterstudiengänge der Fakultäten Geistes- und Kulturwissenschaften, Humanwissenschaften und Sozial- und Wirtschaftswissenschaften an der Otto-Friedrich-Universität Bamberg (vgl. <a href="http://www.uni-bamberg.de/?id=29722">http://www.uni-bamberg.de/?id=29722</a>). Module aus dem Nebenfach Angewandte Informatik sind nicht wählbar.</p> <p>WICHTIG: Die Modulprüfungen einzelner Module eines gewählten Anwendungsfachs müssen inklusive aller erforderlichen zugehörigen Modulteilprüfungen gemäß der gültigen Bestimmungen des gewählten Anwendungsfachs abgelegt werden. Die von der anbietenden Fakultät in der Studien- und Fachprüfungsordnung und im jeweiligen Modulhandbuch festgelegten Regelungen zur Anmeldung und Anrechnung von Modulprüfungen einzelner Module und ihrer zugeordneten Lehrveranstaltungen sind bindend. Im Hinblick auf einzelne Anwendungsfächer gelten oft Zugangsvoraussetzungen, so sind zum Teil Vorpraktika erforderlich. Zu den Details können die Informationen des jeweiligen Anwendungsfachs und die Informationsangebote der dortigen Fachstudienberatung herangezogen werden, vgl: <a href="http://www.uni-bamberg.de/?id=29722">http://www.uni-bamberg.de/?id=29722</a> . Mit der ersten Prüfungsanmeldung zu einem Modul verpflichtet sich der Studierende sowohl zur Beachtung der Zuordnungsregelungen von Lehrveranstaltungen und Modulteilprüfungen des gewählten Moduls als auch der Anerkennung der Prüfungsanmeldungs- und -anrechnungsregelungen des Moduls in dem gewählten Anwendungsfach.</p>				

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
<b>A6 Kontextstudium</b>			<b>12 - 16</b>		
<b>Teil-Modulgruppe: Allgemeine Schlüsselqualifikationen</b>			<b>0 - 16</b>		
EESYS-IITP-B	Internationales IT-Projektmanagement	SS, jährlich	6	2 Vorlesung 2 Übung	schriftliche Prüfung (Klausur) 90 Minuten
ISM-PT-B	Methoden der Präsentation, Gesprächsführung und Diskussion	WS, jährlich	3	2 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
ISM-TA-B	Technikfolgenabschätzung/-bewertung	SS, jährlich	3	2 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
SWT-SSP-B	Soft Skills in IT-Projekten	SS, jährlich	3	2 Vorlesung und Übung	schriftliche Prüfung (Klausur) 90 Minuten
<b>Teil-Modulgruppe: Wissenschaftliches Arbeiten</b>			<b>0 - 3</b>		
MI-WAIAI-B	Einführung in das wissenschaftliche Arbeiten für Informatik und Angewandte Informatik	SS, jährlich(1)	3	2 Vorlesung und Übung	schriftliche Modulprüfung (Klausur) 60 Minuten
<b>Wahlpflichtbereich: Fremdsprachen</b>			<b>0 - 16</b>		
Module gemäß des aktuellen Angebots des Sprachenzentrums, insb. IT English.					
<b>Teil-Modulgruppe: Philosophie / Ethik</b>			<b>0 - 16</b>		
Wählbar sind Module der Philosophie, die dem Studium Generale zugeordnet sind, sowie das folgende Modul:					
PSI-EDS-B	Ethics for the Digital Society	WS, jährlich	3	2 Vorlesung	schriftliche Prüfung (Klausur) 60 Minuten

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
<b>A7 Seminare und Projekte</b>			<b>18</b>		
<p>Es sind zwei Seminarmodule in den Themenbereichen A2-A4 mit jeweils 3 ECTS-Punkten sowie ein Projektmodul Software Systems Science mit 12 ECTS-Punkten zu absolvieren. Mindestens ein Seminarmodul sowie das Projektmodul muss der Fächergruppe Informatik entstammen.</p> <p>Die Zulassung zur Modulprüfung setzt bei den Modulen dieser Modulgruppe eine regelmäßige Teilnahme an den zugehörigen Lehrveranstaltungen gem. § 9 Abs. 10 APO voraus.</p>					
<b>Wahlpflichtbereich: Seminare</b>			<b>6</b>		
<b>Fach: Seminar(e) in Informatik</b>			<b>3 - 6</b>		
DSG-Sem-B	Bachelorseminar zur Praktischen Informatik	WS, SS	3	2 Proseminar	Hausarbeit mit Referat 4 Monate 20 Minuten
Gdl-Sem-B	Bachelorseminar Grundlagen der Informatik	jährlich nach Bedarf WS und SS	3	2 Seminar	Hausarbeit mit Referat 4 Monate 30 Minuten
KTR-Sem-B	Bachelorseminar zu Kommunikationssystemen und Rechnernetzen	jährlich nach Bedarf WS und SS(Regelturnus: SS)	3	2 Seminar	Hausarbeit mit Referat 4 Monate 30 Minuten
MOBI-SEM-B	Bachelor-Seminar Mobile Software Systems	WS, jährlich(1)	3	2 Seminar	Hausarbeit mit Referat
PSI-Sem-B	Seminar Security and Privacy Foundations	SS, jährlich(1)	3	2 Seminar	Hausarbeit mit Referat 3 Monate 30 Minuten
SWT-SEM-B	Seminar Software Engineering and Programming Languages (Bachelor)	WS, SS	3	2 Seminar	Hausarbeit mit Referat 8 Wochen 40 Minuten
<b>Fach: Seminar in Angewandter Informatik oder Wirtschaftsinformatik</b>			<b>0 - 3</b>		
HCI-Sem-B	Bachelorseminar Mensch-Computer-Interaktion	SS, jährlich	3	2 Seminar	Hausarbeit mit Referat

## Modultabelle

						4 Monate 30 Minuten
KInf-Seminar-B	Bachelorseminar Kulturinformatik	SS, jährlich	3	2 Seminar		Hausarbeit mit Referat 20 Minuten
KogSys-Sem-B	Bachelorseminar Kognitive Systeme	WS, jährlich	3	2 Seminar		4 Monate Hausarbeit mit Referat 4 Monate 30 Minuten
MI-Sem-B	Bachelorseminar zur Medieninformatik	SS, jährlich	3	2 Proseminar		Hausarbeit mit Referat 30 Minuten
SME-Sem-B	Bachelorseminar zu Smart Environments	WS, jährlich(1)	3	2 Seminar		Referat mit schriftl. Hausarbeit 4 Monate 30 Minuten
WI-Seminar-B	Bachelorseminar aus der Fächergruppen Wirtschaftsinformatik	WS, SS	3	2 Seminar		Hausarbeit mit Referat
<b>Wahlpflichtbereich: Projekte</b>						
Gdl-PR2-B	Bachelorprojekt Software Systems Science	WS, SS(Beginn jedes Semeste)	12	8 Übung		Hausarbeit mit Kolloquium 4 Monate 30 Minuten schriftliche Hausarbeit 4 Monate
DSG-Project-2- SoSySc-B	DSG Bachelorprojekt Software Systems Science	WS, jährlich	12	8 Übung		schriftliche Hausarbeit 4 Monate Hausarbeit mit Kolloquium 4 Monate 10 Minuten
KTR-SSSProj-B	KTR Bachelorprojekt Software Systems Science	WS, SS(Turnusbeginn SS)	12	8 Bachelorprojekt		schriftliche Hausarbeit 4 Monate Hausarbeit mit Kolloquium 4 Monate 30 Minuten

## Modultabelle

---

SWT-PR2-B	SWT Bachelorprojekt Software Systems Science	WS, SS	12	8 Übung	Hausarbeit mit Kolloquium 12 Wochen 30 Minuten schriftliche Hausarbeit 12 Wochen
MOBI-PRS-B	Bachelor Project Mobile Software Systems (SoSySc)	WS, SS(1)	12	8 Übung	Hausarbeit mit Kolloquium
PSI-SSSProject-B	Software Systems Science Project: Security and Privacy	WS, SS(1)	12	8 Übung	schriftliche Hausarbeit 3 Monate Hausarbeit mit Kolloquium 3 Monate 30 Minuten

## Modultabelle

ID	Modul	Semester	ECTS	SWS	Prüfung
	<b>A8 Bachelorarbeit</b>		<b>12</b>		
SoSySc-Thesis-B	Bachelorarbeit in Software Systems Science	WS, SS(1)	12		schriftliche Hausarbeit 4 Monate