

Otto-Friedrich-Universität  
Bamberg



---

# Modulhandbuch

**Bachelor Software Systems Science**

**Fakultät Wirtschaftsinformatik  
und Angewandte Informatik**

Stand: Wintersemester 2012/2013

Informationen im Web unter <http://www.uni-bamberg.de/wiai/studium/>

---



---

## Module

DSG-AJP-B: Fortgeschrittene Java-Programmierung	3
DSG-Eidl-B: Einführung in die Informatik	5
DSG-EiDistrSys: Einführung in Verteilte Systeme	10
DSG-PKS-B: Programmierung komplexer interagierender Systeme	13
DSG-Sem-B: Bachelorseminar zur Praktischen Informatik	15
Gdl-GTI-B: Grundlagen der Theoretischen Informatik (Machines and Languages)	17
Gdl-Mfl-1: Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik)	20
Gdl-NPP-B: Nichtprozedurale Programmierung	23
Gdl-SaV-B: Logik (Specification and Verification)	26
Gdl-Sem-B: Bachelorseminar Grundlagen der Informatik	28
HCI-IS-B: Interaktive Systeme	30
HCI-KS-B: Kooperative Systeme	33
HCI-US: Ubiquitäre Systeme	36
IAI-WAI-B: Wissenschaftliches Arbeiten in der Informatik	39
KInf-GeoInf-B: Geoinformationssysteme	42
KogSys-IA-B: Intelligent Agents	44
KTR-Datkomm-B: Datenkommunikation	46
KTR-GIK-M: Grundbausteine der Internet-Kommunikation	50
KTR-Mfl-2: Mathematik für Informatiker 2 (Lineare Algebra)	54
KTR-Sem-B: Bachelorseminar	56
Mathe-B-01: Mathematik für Wirtschaftswissenschaftler I (Analysis)	58
MI-AuD-B: Algorithmen und Datenstrukturen	61
MI-LA-DatSchu-B: Grundlagen und Fallstudien zum Datenschutz	64
MI-WebT-B: Web-Technologien	66
SEDA-DMS-B: Datenmanagementsysteme	69
SSS-AIS-B: Bachelorseminar aus der Angewandten Informatik	72
SSS-BAP-B: Bachelorprojekt Software Systems Science	73
Stat-B-01: Methoden der Statistik I	76
Stat-B-02: Methoden der Statistik II	78
SWT-CCP-M: Compiler Construction Project	80
SWT-FSA-B: Foundations of Software Analysis (ab SoSe 2014)	82
SWT-IPC-B: Imperative Programming Using C	85

---

SWT-PCC-M: Principles of Compiler Construction	87
SWT-PMI-B: Projektmanagement in IT-Projekten	90
SWT-PMS-B: Modelle, Methoden und Werkzeuge für das Projektmanagement in Softwareprojekten	92
SWT-SWE-B: Software Engineering	95
SWT-SWL-B: Software Engineering Lab	98
SWT-TPL-B: Trends in Programming Languages (Bachelor)	101
SWT-TSE-B: Trends in Software Engineering (Bachelor)	103

---

# Modul DSG-AJP-B: Fortgeschrittene Java-Programmierung

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3W-I Wahlpflichtbereich I Schwerpunkt PuSWE
<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>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Programmierkenntnisse in Java sowie Grundkenntnisse aus dem Bereich der Algorithmik und Softwareentwicklung, wie sie z.B. im Modul DSG-EidI-B (insbesondere im Teil DSG-EiAPS-B) vermittelt werden.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>Bedingung für ECTS- Punkte</b>	Bearbeitung von kleineren Programmieraufgaben im Semester und mündliches Abschlusskolloquium zu den Aufgaben zum Semesterende.  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 von Programmieraufgaben/assignments</li> <li>• 12.5 Std. Vorbereitung auf das Abschlusskolloquium</li> </ul>
<b>ECTS-Punkte</b>	3

## ***Lehrveranstaltung Praktische Übung Fortgeschrittene Java-Programmierung***

<b>Inhalte</b>	Aufbauend auf den Grundkenntnissen der objekt-orientierten Programmierung in Java aus DSG-EidI 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> </ul>
----------------	--

- Nutzung komplexer Java-APIs, z.B. für Ein- und Ausgabe,
- grundlegende XML Verarbeitung,
- Debugging, Profiling und Testen,
- Überblick über das Programmieren von (grafischen) Benutzerschnittstellen (G)UIs.

Zusätzlich werden die ersten Schritte zur Nutzung komplexer Programmierumgebungen, die über den einfachen Editor-Compiler-Ausführungs-Zyklus hinausgehen, insbesondere der Umgang mit einfachen Testszenerarien zur Entwicklung verlässlicher Systeme, eingeübt.

<b>Dozenten</b>	Mitarbeiter Praktische Informatik
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung und Übung (V/Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Jedes weiterführende Buch zu Java ist verwendbar.
<b>Prüfungen</b>	Abschlusskolloquium zu DSG-AJP-B

### ***Prüfung Abschlusskolloquium zu DSG-AJP-B***

<b>Beschreibung</b>	Während des Semesters regelmäßig ausgegebene Programmieraufgaben (Assignments) werden 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 den semesterbegleitend ausgegebenen Programmieraufgaben vor, erläutern diese und beantworten Fragen zu den vorgestellten Lösungen und den dabei verwendeten Techniken und Sprachkonstrukten.
<b>Typ</b>	Hausarbeit, Kolloquium
<b>Dauer</b>	10 Minuten

# Modul DSGVO-Eidl-B: Einführung in die Informatik

<b>Modulgruppen</b>	A2 Allgemeine Informatik Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Studierende sollen einen ersten Überblick über die verschiedenen Gebiete der Informatik haben und die grundlegenden Begriffe und Methoden der Informatik wie die wichtigsten in der Informatik verwendeten Techniken sowohl aus Sicht der Algorithmen und Softwareentwicklung als auch aus Sicht der 'Informatik der Systeme' kennen.</p> <p>Auf Softwareentwicklungsseite sollen Studierende in der Lage sein, geeignete Abstraktions- und Repräsentationsmethoden auszuwählen, Methoden zur Beschreibung von Syntax und Semantik einfacher Sprachen anzuwenden, die Zusammenhänge zwischen Spezifikation und Implementierung zu verstehen sowie die Arbeitsweise einer Programmiersprache wie auch die wesentlichen Schritte der Softwareentwicklung nachzuvollziehen. Studierende sollen in der Lage sein, einfache Problemstellungen zu beschreiben, algorithmische Lösungen dazu zu entwickeln und diese auch in Java mittels einfacher Datenstrukturen umzusetzen.</p> <p>Auf Systemseite sollen die Studierenden ein grundlegendes Verständnis zustandsbasierter Systeme und der darin möglichen Abläufe haben. Zusätzlich kennen Studierende den grundlegenden Aufbau moderner Rechner- und Betriebssysteme und die dabei zur Anwendung kommenden Informatiktechniken.</p>
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	270 Stunden
<b>Empfohlene Vorkenntnisse</b>	Die Veranstaltung hat als grundlegende Einführungsveranstaltung in das Gebiet der Informatik keine anderen Lehrveranstaltungen oder Programmierkenntnisse zur Voraussetzung.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	<p>Bestehen der zugeordneten Klausur nach Absolvieren der beiden Vorlesungen durch Erreichen von 50% der maximal erreichbaren Punkte.</p> <p>Der Arbeitsaufwand von 270 Std. verteilt sich - bis auf die Klausurvorbereitung - gleichmäßig auf die beiden Semester und gliedert sich in etwa in 115+115+40 Std. also je Semester:</p> <ul style="list-style-type: none"> <li>• 22.5 Std. Vorlesungsteilnahme</li> </ul>

- 22.5 Std. Übungsteilnahme
- 45 Std. Bearbeiten von wöchentlichen Übungsaufgaben
- 25 Std. Vor- und Nachbereitung (Literatur, Recherchen usw.) von Vorlesung und Übung (ohne Bearbeiten der Übungsaufgaben)

Hinzu kommen 40 Std. Vorbereitung auf die Klausur.

**ECTS-Punkte**

9

**Bemerkung**

**Das Modul kann sowohl im Winter- als auch im Sommersemester begonnen werden, da die beiden Vorlesungen nicht aufeinander aufbauen, sondern jeweils einen ersten komplementären Einblick in die Informatik aus Software- sowie aus System-Sicht geben.** Die Modulprüfung kann erst nach 2 Semestern abgelegt werden, da in diesem Modul Inhalte über 2 Semester vermittelt und geprüft werden.

***Lehrveranstaltung DSG-EiAPS: Einführung in Algorithmen, Programmierung und Software***

**Inhalte**

Die Vorlesung DSG-EiAPS gibt einen ersten Einblick in die Informatik aus Sicht der Entwicklung von Algorithmen und deren Realisierung durch Programme in einer imperativen, objekt-orientierten Programmiersprache (Java) sowie einen Ausblick auf die Problematik der Softwareentwicklung. Behandelt werden die Grundprinzipien der Informatik zu:

- Präsentation, Interpretation and Manipulation von Information,
- Syntax and Semantik von einfachen Sprachen,
- Probleme, Problemklassen und -Instanzen,
- Design, Entwicklung und Implementierung von Algorithmen für einfache Problemklassen,
- einfache Datenstrukturen wie Keller, Warteschlangen, Listen und Bäume,
- Techniken zur Spezifikation, zur Datenabstraktion und funktionalen Abstraktion.

All diese Begriffe werden am Beispiel der Programmiersprache 'Java' diskutiert, so dass auch die wesentlichen Konzepte imperativer und objekt-orientierter Programmiersprachen wie

- Wertebereiche, Namensräume, Speichermodelle und Zuweisungen,
- Kontroll- und Datenfluss in einem Programm, sowie
- Klassen, Schnittstellen, Vererbung und Polymorphie

besprochen und auch praktisch eingeübt werden.



<b>Dozenten</b>	Prof. Dr. Guido Wirtz
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<p>Jede Einführung in die Informatik oder in die Programmiersprache Java 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:</p> <ul style="list-style-type: none"><li>• Heinz-Peter Gumm, Manfred Sommer: Einführung in die Informatik. Oldenbourg Verlag, 2011 (9th)</li><li>• Barbara Liskov with John Guttag: Program Development in Java. Addison-Wesley, 2001</li><li>• Timothy Budd: An Introduction to Object-Oriented Programming, Pearson/Addison Wesley, 2002 (3rd)</li><li>• Christian Ullenboom: Java ist auch eine Insel. Galileo Computing, 2011 (9th)</li><li>• John Lewis, Joseph Chase: Java Software Structures. Pearson/Addison-Wesley, 2010 (3rd)</li><li>• C. Heinisch, F. Müller, J. Goll: Java als erste Programmiersprache. Teubner, 2007 (5th)</li></ul>

**Prüfungen** Klausur zu DSG-EidI-B

### ***Lehrveranstaltung DSG-EiAPS Übung***

**Inhalte** In der Übung werden die wichtigsten Konzepte der gleichnamigen Vorlesung an einfachen Beispielen praktisch umgesetzt und durch die Besprechung von regelmäßig zu lösenden (unbenoteten) Hausaufgaben vertieft. Dabei wird insbesondere Wert auf die Vorstellung von Lösungen durch die Studierenden und deren Diskussion in der Übungsgruppe gelegt.

<b>Dozenten</b>	Mitarbeiter Praktische Informatik
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	- vgl. Vorlesung -

**Prüfungen**

Klausur zu DSG-EidI-B

***Lehrveranstaltung DSG-EiRBS: Einführung in Rechner- und Betriebssysteme***

**Inhalte**

Die Vorlesung 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 von Rechner- und Betriebssystemen. Sie bietet einen Einblick in Aufbau und Architektur monolithischer Rechnersysteme. Dazu gehört neben dem schrittweisen Aufbau eines minimalen Rechners, beginnend mit aussagenlogischen Ausdrücken über ihre Realisierung durch Gatter und Standardbausteine sowie zustandsbehaftete Schaltungen und Speicherbausteinen auch die Darstellung von Daten im Rechner und ihre detaillierte Speicherung und Verarbeitung. Zusätzlich wird ein Überblick über das Zusammenspiel von Konzepten der Rechnerarchitektur mit den wichtigsten Prinzipien und Komponenten von Systemsoftware (Prozess- und Ressource-Scheduling, Speicherverwaltung, Hintergrundspeicher, I/O-Handhabung) gegeben. Die Vorlesung gibt zusätzlich einen Ausblick auf moderne Techniken der Prozessorarchitektur und Multiprozessorarchitekturen, wie sie in aktuellen Serverkonstellationen zum Einsatz kommen. Die Themen werden anhand von Modellen sowie anhand von marktgängigen Rechner- und Betriebssystemen behandelt.

**Dozenten**

Prof. Dr. Guido Wirtz

**Sprache**

Deutsch

**Lehrformen**

Vorlesung (V)

**Häufigkeit**

SS, jährlich

**SWS**

2

**Literatur**

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 auch nur eins dieser Bücher erfolgreich absolviert werden.

- Tanenbaum, A.S./Goodman J.: Computerarchitektur. Pearson Studium/Prentice Hall, 2005 (5th)
- Murdocca, M./Heuring, V.P.: Computer Architecture and Organization. Prentice Hall 2007 (1th)

- Tanenbaum, A.S.: Moderne Betriebssysteme. Pearson Studium 2009 (3rd)
- Silberschatz, A./Gagne, G./Galvin, P. B.: Operating Systems Concepts. John Wiley and Sons, 2010 (8th)

**Prüfungen** Klausur zu DSG-EidI-B

### ***Lehrveranstaltung DSG-EiRBS Übung***

**Inhalte** In der Übung werden die wichtigsten Konzepte der gleichnamigen Vorlesung an einfachen Beispielen praktisch umgesetzt und durch die Besprechung von regelmäßig zu lösenden (unbenoteten) Hausaufgaben vertieft. Dabei wird insbesondere Wert auf die Vorstellung von Lösungen durch die Studierenden und deren Diskussion in der Übungsgruppe gelegt.

**Dozenten** Mitarbeiter Praktische Informatik

**Sprache** Deutsch

**Lehrformen** Übung (Ü)

**Häufigkeit** SS, jährlich

**SWS** 2

**Literatur** - vgl. Vorlesung -

**Prüfungen** Klausur zu DSG-EidI-B

### ***Prüfung Klausur zu DSG-EidI-B***

**Beschreibung** 120-minütige Klausur zum Stoff des gesamten Moduls, also der Vorlesungen und Übungen zu DSG-EiAPS und DSG-EiRBS.

**Typ** Klausur

**Dauer** 120 Minuten

## Modul DSG-EiDistrSys: Einführung in Verteilte Systeme

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4P Pflichtbereich Grundlagen KuVS
<b>Lernziele / Kompetenzen</b>	Studierende sollen die typischen Charakteristiken moderner verteilter und mobiler Systeme, die grundlegenden Unterschiede zu klassischen monolithischen Systemen und die sich dadurch ergebenden Vor- und Nachteile kennen und die derzeit gängigen konzeptionellen, algorithmischen und programmiersprachlichen Techniken zur Realisierung robuster und sicher benutzbarer verteilter Systeme auch praktisch zur Umsetzung einfacher verteilter Systeme anwenden können.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Allgemeine Grundlagen der Informatik mit einem Schwerpunkt auf systemnaher Informatik (insbesondere Betriebssysteme) und praktischer Programmierung in Java, vorzugsweise auch parallele Programmierung in Java mit Threads und Synchronisationskonstrukten.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>Bedingung für ECTS- Punkte</b>	Bearbeitung von Assignments während des Semesters sowie Bestehen des mündlichen Abschlusskolloquiums. Das Abschlusskolloquium überprüft sowohl die erreichten Kenntnisse aus den Assignments als auch das Verständnis der in der Veranstaltung vermittelten Grundsätze, Konzepte und Technologien.  Der Arbeitsaufwand von insgesamt 180 Std. gliedert sich in etwa in: <ul style="list-style-type: none"><li>• 22.5 Std. Vorlesungsteilnahme</li><li>• 22.5 Std. Übungsteilnahme und Vorstellung/Besprechung der Assignments</li><li>• 90 Std. Bearbeiten der Programmier-Assignments über das Semester verteilt</li><li>• 25 Std. Vor- und Nachbereitung (Literatur, Recherchen usw.) der Vorlesung (ohne Bearbeiten der Assignments)</li><li>• 20 Std. Vorbereitung auf das Kolloquium</li></ul>
<b>ECTS-Punkte</b>	6

***Lehrveranstaltung Vorlesung Einführung in Verteilte Systeme***

**Inhalte** Die Veranstaltung gibt eine Einführung in das Gebiet verteilter und mobiler Systeme. Sie beschäftigt sich mit der Charakterisierung und Anwendung verteilter und mobiler Systeme und ihren konzeptionellen und technologischen Grundlagen auf Netzwerk-, Betriebssystem-, Programmiersprachen- und Middleware-Ebene. Dabei spielen alternative Interaktions-Paradigmen, das damit verbundene Maß an Kopplung und Abhängigkeit zwischen Teilsystemen und die jeweilige Bewertung im Kontext verteilter und mobiler Systeme eine zentrale Rolle. Neben den konzeptionellen Fragen werden auch praktische Erfahrungen mit den entsprechenden Programmierparadigmen wie z.B. Messaging oder Transaktionsverarbeitung vermittelt.

Zusätzlich werden die wichtigsten Klassen verteilter Algorithmen sowie Techniken zur Implementierung von Leistungs- und Ausfall-Transparenz als Grundlage verlässlicher Systeme diskutiert und auf ihre praktische Verwendung hin analysiert.

**Dozenten** Prof. Dr. Guido Wirtz

**Sprache** Deutsch

**Lehrformen** Vorlesung (V)

**Häufigkeit** SS, jährlich

**SWS** 2

**Literatur**

- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair: Distributed Systems. Pearson Education UK 2011 (5th); ISBN: 9780273760597; 1072 Seiten
- Andrew S. Tanenbaum, Maarten van Steen: Verteilte Systeme - Prinzipien und Paradigmen. Pearson Studium 2007 (2. Auflage); ISBN: 978-3-8273-7293-2; 768 Seiten

**Prüfungen** Prüfung Einführung in Verteilte Systeme

### ***Lehrveranstaltung Übung zur Einführung in verteilte Systeme***

**Inhalte** In der Übung werden die regelmäßig zu praktischen Aspekten der Vorlesung in Gruppen zu bearbeitenden Assignments diskutiert und auftretende technische Probleme gelöst. Dabei wird insbesondere Wert auf die Vorstellung von Lösungen durch die Studierenden und deren Vergleich in der Übungsgruppe gelegt.

**Dozenten** Mitarbeiter Praktische Informatik

**Sprache** Deutsch

**Lehrformen** Übung (Ü)

**Häufigkeit** SS, jährlich

**SWS** 2

**Literatur** - siehe gleichnamige Vorlesung -

**Prüfungen** Prüfung Einführung in Verteilte Systeme

***Prüfung Prüfung Einführung in Verteilte Systeme***

**Beschreibung** Mündliches Prüfungsgespräch über die Inhalte von Vorlesung, Übung und Assignments.

**Typ** Hausarbeit, Kolloquium

**Dauer** 20 Minuten

# Modul DSG-PKS-B: Programmierung komplexer interagierender Systeme

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4P Pflichtbereich Grundlagen KuVS
<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>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Programmierkenntnisse in Java sowie Grundkenntnisse aus dem Bereich der Betriebssysteme, wie sie z.B. im Modul DSG-EidI-B (insbesondere im Teil DSG-EiRBS-B) vermittelt werden.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>Bedingung für ECTS- Punkte</b>	Bearbeitung von kleineren Programmieraufgaben im Semester ('schriftliche Hausarbeit') und mündliches Abschlusskolloquium zu den Aufgaben zum Semesterende.  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 von Programmieraufgaben</li> <li>• 12.5 Std. Vorbereitung auf das Abschlusskolloquium (unter Voraussetzung der schon erbrachten o.g. Aufwände !)</li> </ul>
<b>ECTS-Punkte</b>	3

## ***Lehrveranstaltung Praktische Übung Programmierung komplexer interagierender Systeme***

<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

- Prozesse und Threads,
- Prozesskommunikation: Shared Memory vs. Message Passing Paradigmen,
- Synchronisation bei Shared Memory,
- Message Passing mit Sockets,
- einfache C/S-Systeme und Remote Procedure Calls am Bsp. Java RMI.

Zusätzlich wird die Problematik robuster verteilter Systeme diskutiert und ein Ausblick auf alternative Interaktionsparadigmen gegeben.

<b>Dozenten</b>	Mitarbeiter Praktische Informatik
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung und Übung (V/Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	- wird jeweils aktuell zur Veranstaltung angegeben -
<b>Prüfungen</b>	Abschlusskolloquium zu DSG-PKS-B

### ***Prüfung Abschlusskolloquium zu DSG-PKS-B***

<b>Beschreibung</b>	Während des Semesters regelmäßig ausgegebene Programmieraufgaben (Assignments) werden 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 den semesterbegleitend ausgegebenen Programmieraufgaben vor, erläutern diese und beantworten Fragen zu den vorgestellten Lösungen und den dabei verwendeten Techniken und Sprachkonstrukten.
<b>Typ</b>	Hausarbeit, Kolloquium
<b>Dauer</b>	10 Minuten



# Modul DSGVO-Sem-B: Bachelorseminar zur Praktischen Informatik

<b>Modulgruppen</b>	A7 Seminare und Projekte->Seminare
<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>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse im jeweils im Seminar behandelten Gebiet der Praktischen Informatik, also mindestens eine der beiden Veranstaltungen aus dem Modul DSGVO-EidI-B.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSGV-EidI-B)
<b>Bedingung für ECTS- Punkte</b>	Erstellen einer schriftlichen Hausarbeit auf der Grundlage einer Literaturrecherche zum gestellten Thema, Einhalten der Besprechungstermine für Gliederung, Hausarbeit und Referat sowie regelmäßige aktive Teilnahme an den Diskussionen im Seminar. Der Arbeitsaufwand von 90 Std. gliedert sich in etwa in: <ul style="list-style-type: none"> <li>• 20 Std. Besprechungen und Vorträge mit Diskussion</li> <li>• 25 Std. Literaturrecherche sowie Erarbeitung und Bewertung der Literatur</li> <li>• 30 Std. Anfertigen der schriftlichen Ausarbeitung (Hausarbeit)</li> <li>• 15 Std. Vorbereitung des Referats (Vortrag)</li> </ul>
<b>ECTS-Punkte</b>	3
<b><i>Lehrveranstaltung Bachelorseminar zur Praktischen Informatik</i></b>	
<b>Inhalte</b>	Verschiedene Themen aus dem Bereich der praktischen Informatik, die einen der fachlichen oder methodischen Aspekte aus dem grundlegenden Informatik-Modul DSGVO-EidI (Einführung in die Informatik) anhand aktueller Literatur vertiefen und/oder ergänzen.
<b>Dozenten</b>	Mitarbeiter Praktische Informatik Prof. Dr. Guido Wirtz
<b>Sprache</b>	Deutsch

<b>Lehrformen</b>	Seminar (S)
<b>Häufigkeit</b>	WS, SS
<b>SWS</b>	2
<b>Literatur</b>	- wird jeweils nach Seminarthemen vergeben -
<b>Prüfungen</b>	Ausarbeitung und Seminarvortrag

***Prüfung Ausarbeitung und Seminarvortrag***

**Beschreibung** Begutachtung einer schriftlichen Ausarbeitung zu den wichtigsten Aspekten des erarbeiteten Themas mit formgerechter Liste der verwendeten Literatur. Freies Halten eines Referats auf der Grundlage der von dem/der Vortragenden erstellten Folien oder elektronischen Präsentationsunterlagen inklusive Diskussion der Inhalte mit den SeminarteilnehmerInnen.

**Typ** Hausarbeit, Referat

**Dauer** 30 Minuten

# Modul Gdl-GTI-B: Grundlagen der Theoretischen Informatik (Machines and Languages)

<b>Modulgruppen</b>	A2 Allgemeine Informatik Grundlagen
<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>WWW</b>	<a href="http://www.gdi.uni-bamberg.de/">http://www.gdi.uni-bamberg.de/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Englischkenntnisse.
<b>Inhaltlich</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>vorausgesetzte Module</b>	Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (Gdl-MfI-1)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur  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): 15 Stunden</li> <li>• Prüfungsvorbereitung: 30 Stunden</li> </ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	

***Lehrveranstaltung Übung Grundlagen der Theoretischen Informatik***

<b>Inhalte</b>	Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung. Sie dient damit auch der Klausurvorbereitung.
<b>Dozenten</b>	Prof. Ph.D. Michael Mendler N.N.
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	
<b>Prüfungen</b>	Grundlagen der Theoretischen Informatik

### ***Lehrveranstaltung Vorlesung Grundlagen der Theoretischen Informatik***

<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 sowie der Lambda-Kalkül als Basis zum Verständnis funktionaler und anderer höherer Programmiersprachen stehen 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, etwa die Chomsky Hierarchie und die P/NP Komplexitätsklassen, besprochen. Über die klassischen Modelle der Algorithmentheorie hinaus werden, je nach verfügbarer Zeit, auch neuere Semantiken für nebenläufige und verteilte sowie objektorientierte Programmierung eingeführt und an Beispielen diskutiert.
<b>Dozenten</b>	Prof. Ph.D. Michael Mendler
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2

**Literatur**

- Hopcroft, J. E., Motwani, R., Ullman, J. D.: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, Pearson Studium, 2002.
- Asteroth, A., Baier, Ch.: Theoretische Informatik, Pearson Studium, 2002.
- Martin, J. C.: Introduction to Languages and the Theory of Computation, McGraw Hill, (2nd ed.),1997.
- Milner, R.: Communicating and Mobile Systems: the p-Calculus. Cambridge University Press, 1999.

**Prüfungen**

Grundlagen der Theoretischen Informatik

***Prüfung Grundlagen der Theoretischen Informatik*****Beschreibung****Typ**

Klausur

**Dauer**

90 Minuten

## **Modul Gdl-Mfl-1: Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik)**

<b>Modulgruppen</b>	A1 Mathematische Grundlagen
<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>WWW</b>	<a href="http://www.gdi.uni-bamberg.de/">http://www.gdi.uni-bamberg.de/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Englischkenntnisse
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur
	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 Teilnahme an Rechnerübungen: 45 Stunden</li><li>• Prüfungsvorbereitung: 30 Stunden</li></ul>
<b>ECTS-Punkte</b>	6

### ***Lehrveranstaltung Vorlesung Mathematik für Informatiker***

<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>Dozenten</b>	Prof. Ph.D. Michael Mendler
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<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>
<b>Prüfungen</b>	Mathematik für Informatiker

### ***Lehrveranstaltung Übung Mathematik für Informatiker***

<b>Inhalte</b>	Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung. Sie dient damit auch der Klausurvorbereitung.
<b>Dozenten</b>	Prof. Ph.D. Michael Mendler N. N.
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	-
<b>Prüfungen</b>	Mathematik für Informatiker
<b><i>Prüfung Mathematik für Informatiker</i></b>	
<b>Typ</b>	Klausur

**Dauer**

90 Minuten



# Modul GdI-NPP-B: Nichtprozedurale Programmierung

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3P Pflichtbereich Grundlagen PuSWT
<b>Lernziele / Kompetenzen</b>	Fähigkeit zur Entwicklung algorithmischer Problemlösungen in nichtprozeduralen Programmiersprachen; Einsicht in die Bedeutung formaler Semantiken für die Implementierung von Programmiersprachen und die Fähigkeit, die funktionale Korrektheit einfacher Programme über ihre formale Semantik zu verifizieren; Kenntnis verschiedener Techniken zur Semantikgebung, insbesondere die denotationelle, operationelle, und Termersetzungssemantik; die Fähigkeit neue Sprachkonstrukte mit diesen Techniken zu spezifizieren; Fähigkeit, sich neue Programmiersprachen systematisch zu erarbeiten und diese in ihren Anwendungsmöglichkeiten kompetent einzuordnen; Kenntnis deklarativer Modelle interaktiver Software und die Fähigkeit, diese in einer konkreten Programmiersprache zu implementieren.
<b>WWW</b>	<a href="http://www.gdi.uni-bamberg.de/">http://www.gdi.uni-bamberg.de/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Englischkenntnisse
<b>Inhaltlich</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>vorausgesetzte Module</b>	Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (GdI-MfI-1)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur  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 Teilnahme an Rechnerübungen: 45 Stunden</li> <li>• Prüfungsvorbereitung: 30 Stunden</li> </ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	
	<b><i>Lehrveranstaltung Vorlesung Nichtprozedurale Programmierung</i></b>

<b>Inhalte</b>	Die Veranstaltung beschäftigt sich mit den Grundlagen der Logikprogrammierung und der funktionalen Programmierung als die wichtigsten Alternativen 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. An Beispielen wird die deklarative Programmierung interaktiver Anwendungen nach dem synchronen Programmierprinzip (synchrone Kahn-Netzwerke) aufgezeigt.
<b>Dozenten</b>	Prof. Ph.D. Michael Mendler
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<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><li>• O’Keefe, R. A.: The Craft of Prolog. MIT Press, 2nd printing, 1994.</li></ul>
<b>Prüfungen</b>	Nichtprozedurale Programmierung

***Lehrveranstaltung Übung Nichtprozedurale Programmierung***

<b>Inhalte</b>	Die Übung vertieft die Konzepte und Konstruktionen aus der Vorlesung. Sie dient damit auch der Klausurvorbereitung.
<b>Dozenten</b>	Prof. Ph.D. Michael Mendler
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	

**Prüfungen** Nichtprozedurale Programmierung

***Prüfung Nichtprozedurale Programmierung***

**Beschreibung**

**Typ** Klausur

**Dauer** 90 Minuten

## Modul Gdl-SaV-B: Logik (Specification and Verification)

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4W Wahlpflichtbereich Komplexe und Verteilte Systeme
<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>WWW</b>	<a href="http://www.gdi.uni-bamberg.de/">http://www.gdi.uni-bamberg.de/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Englischkenntnisse
<b>Inhaltlich</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>vorausgesetzte Module</b>	Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (GdI-MfI-1)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur  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): 15 Stunden</li><li>• Prüfungsvorbereitung: 30 Stunden</li></ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	
<b><i>Lehrveranstaltung Vorlesung/Übung Logik (Specification and Verificati- on)</i></b>	
<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 *Computational Logic* 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>Dozenten</b>	Prof. Ph.D. Michael Mendler
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Vorlesung und Übung (V/Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	4
<b>Literatur</b>	<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>
<b>Prüfungen</b>	Logik
<b><i>Prüfung Logik</i></b>	
<b>Beschreibung</b>	
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## **Modul GdI-Sem-B: Bachelorseminar Grundlagen der Informatik**

<b>Modulgruppen</b>	A7 Seminare und Projekte->Seminare
<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>WWW</b>	<a href="http://www.gdi.uni-bamberg.de/">http://www.gdi.uni-bamberg.de/</a>
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Mathematik für Informatiker, Einführung in die Informatik, Rechner- und Betriebssysteme, Grundlagen der Theoretischen Informatik, Englischkenntnisse.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Benotete Hausarbeit und Referat (eine Note)  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>ECTS-Punkte</b>	3

### ***Lehrveranstaltung Bachelorseminar Grundlagen der Informatik***

<b>Inhalte</b>	Das GdI-Seminar wird zu semesterweise wechselnden Themen angeboten.
<b>Dozenten</b>	Michael Mendler N.N.
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Seminar (S)
<b>Häufigkeit</b>	jährlich nach Bedarf WS und SS (jährlich nach Bedarf im Winter- und Sommersemester)
<b>SWS</b>	2

**Literatur**                    Literatur wird bei Ankündigung bzw. zu Beginn des Seminars bekanntgegeben.

**Prüfungen**                    schriftliche Hausarbeit (2 Wochen) und Referat

***Prüfung schriftliche Hausarbeit (2 Wochen) und Referat***

**Typ**                              Hausarbeit, Referat

**Dauer**                         30 Minuten

## Modul HCI-IS-B: Interaktive Systeme

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3W-II Wahlpflichtbereich II Schwerpunkt PuSWE
<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>WWW</b>	<a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundkenntnisse in Informatik im Umfang einer Einführung in die Informatik
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur  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 Teilleistungen): ca. 30 Stunden</li><li>• Bearbeiten der 6 Teilleistungen: insgesamt ca. 45 Stunden</li><li>• Prüfungsvorbereitung: ca. 30 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li></ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	

### ***Lehrveranstaltung Vorlesung Interaktive Systeme***

<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><li>• Evaluierung von interaktiven Systemen</li><li>• Entwicklungsprozess interaktiver Systeme</li></ul>
----------------	---



- Interaktive Systeme im größeren Kontext und verwandte Themen

<b>Dozenten</b>	Prof. Dr. Tom Gross
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Die Veranstaltung ist eine Zusammenstellung verschiedener Quellen; als ergänzende Quelle und zum Nachschlagen wird empfohlen: <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>
<b>Prüfungen</b>	Interaktive Systeme (Klausur)

### ***Lehrveranstaltung Übung Interaktive Systeme***

<b>Inhalte</b>	praktische Aufgaben zum Vorlesungsstoff einschließlich der Programmierung kleiner Prototypen
<b>Dozenten</b>	Mitarbeiter Mensch-Computer-Interaktion
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung
<b>Prüfungen</b>	Interaktive Systeme (Klausur)

### ***Prüfung Interaktive Systeme (Klausur)***

<b>Beschreibung</b>	In der <b>Klausur</b> können 90 Punkte erzielt werden.  Im Semester werden darüber hinaus 6 <b>Teilleistungen</b> zur Bearbeitung ausgegeben. Für jede Teilleistung stehen in der Regel 2 Wochen als Bearbeitungszeit zur Verfügung. Die Lösungen zu den Teilleistungen werden bewertet. Pro Teilleistung können maximal 2 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 (also maximal 12 Punkte) als Bonuspunkte angerechnet. Eine 1,0 ist dabei aber auf jeden Fall auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar.

**Typ**

Klausur

**Dauer**

90 Minuten

# Modul HCI-KS-B: Kooperative Systeme

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4W Wahlpflichtbereich Komplexe und Verteilte Systeme
<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>WWW</b>	<a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundkenntnisse in Informatik im Umfang einer Einführung in die Informatik sowie Programmierkenntnisse in Java.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur  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 Teilleistungen): ca. 30 Stunden</li> <li>• Bearbeiten der 6 Teilleistungen: insgesamt ca. 45 Stunden</li> <li>• Prüfungsvorbereitung: ca. 30 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li> </ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	
<b><i>Lehrveranstaltung Vorlesung Kooperative Systeme</i></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>

- Technologische Unterstützung für wechselseitige Information, Kommunikation, Koordination, Gruppenarbeit und Online-Gemeinschaften
- Analyse kooperativer Umgebungen
- Entwurf von CSCW und Groupware
- Implementation von CSCW und Groupware
- CSCW im größeren Kontext und verwandte Themen

<b>Dozenten</b>	Prof. Dr. Tom Gross
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Die Veranstaltung ist eine Zusammenstellung verschiedener Quellen; als ergänzende Quelle und zum Nachschlagen wird empfohlen: <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>
<b>Prüfungen</b>	Kooperative Systeme (Klausur)

### ***Lehrveranstaltung Übung Kooperative Systeme***

<b>Inhalte</b>	praktische Aufgaben zum Vorlesungsstoff einschließlich der Programmierung kleiner Prototypen
<b>Dozenten</b>	Mitarbeiter Mensch-Computer-Interaktion
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung
<b>Prüfungen</b>	Kooperative Systeme (Klausur)

### ***Prüfung Kooperative Systeme (Klausur)***

**Beschreibung** In der **Klausur** können 90 Punkte erzielt werden.

Im Semester werden darüber hinaus 6 **Teilleistungen** zur Bearbeitung ausgegeben. Für jede Teilleistung stehen in der Regel 2 Wochen als Bearbeitungszeit zur Verfügung. Die Lösungen zu den Teilleistungen werden bewertet. Pro Teilleistung können maximal 2 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 (also maximal 12 Punkte) als Bonuspunkte angerechnet. Eine 1,0 ist dabei aber auf jeden Fall auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar.

**Typ**

Klausur

**Dauer**

90 Minuten

## Modul HCI-US: Ubiquitäre Systeme

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4W Wahlpflichtbereich Komplexe und Verteilte Systeme
<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>WWW</b>	<a href="http://www.uni-bamberg.de/hci/leistungen/studium">http://www.uni-bamberg.de/hci/leistungen/studium</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Kenntnisse entsprechend den unter "Notwendige Module" angegebenen Modulen
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSG-EidI-B) Modul Algorithmen und Datenstrukturen (MI-AuD-B)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur  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 6 Teilleistungen): ca. 75 Stunden</li><li>• Prüfungsvorbereitung: ca. 30 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li></ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	

### ***Lehrveranstaltung Vorlesung Ubiquitäre Systeme***

<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 und miteinander kommunizieren können - die folgenden Themen konzeptionell, technisch und methodisch behandelt: <ul style="list-style-type: none"><li>• Grundlegende Konzepte</li></ul>
----------------	---

- Basistechnologie und Infrastrukturen
- Ubiquitäre Systeme und Prototypen
- Kontextadaptivität
- Benutzerinteraktion
- Ubiquitäre Systeme im größeren Kontext und verwandte Themen

<b>Dozenten</b>	Prof. Dr. Tom Gross
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Die Veranstaltung ist eine Zusammenstellung verschiedener Quellen; als ergänzende Quelle und zum Nachschlagen wird empfohlen: <ul style="list-style-type: none"> <li>• Krumm, J., (Hrsg.). Ubiquitous Computing Fundamentals. Taylor &amp; Francis Group, Boca Raton, FL, 2010.</li> </ul>
<b>Prüfungen</b>	Ubiquitäre Systeme (Klausur)

### ***Lehrveranstaltung Übung Ubiquitäre Systeme***

<b>Inhalte</b>	praktische Aufgaben zum Vorlesungsstoff einschließlich der Programmierung kleiner Prototypen
<b>Dozenten</b>	Mitarbeiter Mensch-Computer-Interaktion
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung
<b>Prüfungen</b>	Ubiquitäre Systeme (Klausur)

### ***Prüfung Ubiquitäre Systeme (Klausur)***

<b>Beschreibung</b>	In der <b>Klausur</b> können 90 Punkte erzielt werden.  Im Semester werden darüber hinaus 6 <b>Teilleistungen</b> zur Bearbeitung ausgegeben. Für jede Teilleistung stehen in der Regel 2 Wochen als Bearbeitungszeit zur Verfügung. Die Lösungen zu den Teilleistungen werden bewertet. Pro Teilleistung können maximal 2 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 (also maximal 12 Punkte) als Bonuspunkte angerechnet. Eine 1,0 ist dabei aber auf jeden Fall auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar.

**Typ**

Klausur

**Dauer**

90 Minuten



# Modul IAI-WAI-B: Wissenschaftliches Arbeiten in der Informatik

<b>Modulgruppen</b>	A6 Kontextstudium->Wissenschaftliches Arbeiten
<b>Lernziele / Kompetenzen</b>	<p>Dieses Modul richtet sich an Studierende der Studiengänge B.Sc. Angewandte Informatik und B.Sc. Software Systems Science sowie interessierte Studierende anderer Studiengänge, die im Bereich Informatik eine Projekt-, Seminar-, Bachelor- oder Masterarbeit schreiben möchten. Das Modulziel ist, diese Studierenden in zentrale Methoden, Techniken und Werkzeuge des wissenschaftlichen Arbeitens theoretisch und praktisch einzuführen. Nach erfolgreicher Teilnahme an diesem Modul können Studierende insbesondere:</p> <ul style="list-style-type: none"> <li>• wissenschaftliches Arbeiten - einschließlich kollaboratives Arbeiten - planen und managen;</li> <li>• Literaturrecherchen selbständig durchführen und die Güte verschiedener Quellen einschätzen;</li> <li>• den Aufbau einer wissenschaftlichen Arbeit anhand formaler Anforderungen erkennen und beurteilen;</li> <li>• die Gliederung, die Problemstellung und das Literaturverzeichnis einer wissenschaftlichen Arbeit erstellen;</li> <li>• typische Forschungsmethoden in der Informatik einordnen und anwenden;</li> <li>• experimentelle Studien durchführen und auswerten;</li> <li>• wissenschaftliche Vorträge vorbereiten und halten;</li> <li>• elementare Softwarewerkzeuge zur Unterstützung des wissenschaftlichen Arbeitens benutzen.</li> </ul>
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Keine. Der Besuch des Moduls im 1. Semester wird allerdings nicht empfohlen.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung 90 Std., welche sich grob wie folgt gliedern:

- 30 Std. Teilnahme an der Vorlesung/Übung
- 30 Std. Vor- und Nachbereitung der Vorlesung/Übung
- 30 Std. Anfertigen einer schriftlichen Hausarbeit

**ECTS-Punkte** 3

**Bemerkung**

***Lehrveranstaltung Vorlesung/Übung zu Wissenschaftliches Arbeiten in der Informatik***

<b>Inhalte</b>	<p>Die Vorlesung/Übung bietet den Studierenden einen Einblick in zentrale Themenbereiche des wissenschaftlichen Arbeitens:</p> <ul style="list-style-type: none"><li>• Zeit- und Selbstmanagement</li><li>• Literaturrecherche, sowie Literaturverwaltung</li><li>• Verfassen und Strukturieren wissenschaftlicher Arbeiten in deutscher und englischer Sprache (einschließlich richtiges Zitieren)</li><li>• Typische Forschungsmethoden in der Informatik</li><li>• Durchführen und Auswerten experimenteller Studien (einschließlich Benchmarking)</li><li>• Erstellen und Präsentieren von Vorträgen</li><li>• Wissenschaftliche Zusammenarbeit</li><li>• Ethische Aspekte des wissenschaftlichen Arbeitens</li></ul> <p>Begleitend werden diese theoretischen Inhalte anhand einer konkreten Forschungsfrage praktisch eingeübt und vertieft. Dabei werden auch verschiedene Software-Werkzeuge - wie z. B. LaTeX/BibTeX bzw. Citavi, Gnuplot, Matlab und Versionsmanagement-Tools - vorgestellt und von den Studierenden aktiv eingesetzt.</p>
<b>Dozenten</b>	<p>Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen</p>
<b>Sprache</b>	<p>Deutsch/Englisch</p>
<b>Lehrformen</b>	<p>Vorlesung und Übung (V/Ü)</p>
<b>Häufigkeit</b>	<p>jährlich</p>
<b>SWS</b>	<p>2</p>
<b>Literatur</b>	<p>Wird zu Beginn der Veranstaltung bekannt gegeben.</p>
<b>Prüfungen</b>	<p>-</p>

***Prüfung Wissenschaftliches Arbeiten in der Informatik***

<b>Beschreibung</b>	Selbständiges Anwenden der in der Vorlesung/Übung vermittelten Inhalte und Werkzeuge auf eine kleine Fallstudie, deren Ausgangspunkt eine simple Forschungsfrage ist.
<b>Typ</b>	Hausarbeit
<b>Dauer</b>	3 Wochen

## Modul KInf-GeoInf-B: Geoinformationssysteme

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4W Wahlpflichtbereich Komplexe und Verteilte Systeme
<b>Lernziele / Kompetenzen</b>	<ul style="list-style-type: none"><li>• Kenntnis der Grundbegriffe und der informatischen Methoden aus dem Bereich der Geoinformationssysteme</li><li>• Orientierungswissen, das den Methodenvergleich sowie die Zuordnung von Anwendungsproblemen zu geeigneten Methoden ermöglicht</li><li>• Fähigkeit, Methoden auf Problemstellungen anwenden zu können</li><li>• Fähigkeit, Problemstellungen im Team zu analysieren und zu lösen</li><li>• Datenanalyse mit GIS</li></ul>
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Die Inhalte der Veranstaltung "Einführung in die Informatik" oder "Informatik und Programmierkurs für die Kulturwissenschaften" (oder entsprechende Vorkenntnisse) werden vorausgesetzt.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSG-EidI-B) Modul Informatik und Programmierkurs für die Kulturwissenschaften (KInf-IPKult-E)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Abschlussklausur
<b>ECTS-Punkte</b>	6

### **Lehrveranstaltung Vorlesung Geoinformationssysteme**

<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.
----------------	--

---

<b>Dozenten</b>	Prof. Dr. Christoph Schlieder
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<p>Longley, P., Goodchild, M., Maguire, D., Rhind, D. (2001): Geographic Information: Systems and Science, Wiley: Chichester, UK.</p> <p>Shekhar, S., Chawla, S. (2003): Spatial Databases: A Tour, Prentice Hall: Upper Saddle River, NJ.</p> <p>Smith, M., Goodchild, M., and Longley, P. (2007): Geospatial Analysis, 2nd edition, Troubador Publishing Ltd.</p>

**Prüfungen** Geoinformationssysteme (schriftlich)

### ***Lehrveranstaltung Übung Geoinformationssysteme***

<b>Inhalte</b>	siehe Vorlesung
<b>Dozenten</b>	Mitarbeiter Angewandte Informatik in den Kultur-, Geschichts- und Geowissenschaften
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung
<b>Prüfungen</b>	Geoinformationssysteme (schriftlich)

### ***Prüfung Geoinformationssysteme (schriftlich)***

<b>Beschreibung</b>	In der schriftlichen Prüfung werden die in Vorlesung und Übung behandelten Themengebiete geprüft.
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## Modul KogSys-IA-B: Intelligent Agents

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4W Wahlpflichtbereich Komplexe und Verteilte Systeme
<b>Lernziele / Kompetenzen</b>	This modul provides basic knowledge and skills in the area of “cognition-oriented artificial intelligence”. It is focused on problem solving and planning.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/kogsys/teaching/">http://www.uni-bamberg.de/kogsys/teaching/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	See "Vorleistungen", recommended for 4th semester
<b>Inhaltlich vorausgesetzte Module</b>	Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (GdI-MfI-1) Modul Algorithmen und Datenstrukturen (MI-AuD-B)
<b>Bedingung für ECTS- Punkte</b>	Passing the exam. The work for this modul can coarsely be divided as follows: 22.5h lectures + 40h studying at home (over 15 weeks) 22.5h practices + 60h assignments (over 15 weeks) 30h exam preparation
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	If all students prefer German as teaching language, the oral presentations, the assignments and the exam will be given in German.  Slides and textbooks are English in every case.

### ***Lehrveranstaltung Lecture Intelligent Agents***

<b>Inhalte</b>	Basic concepts and methods of cognition-oriented artificial intelligence are introduced. The main focus is on problem solving and planning. Main topics are: STRIPS-planning, logic and deductive planning, heuristic search and planning, graph plan algorithms, multiagent planning, references to human problem solving and planning.
<b>Dozenten</b>	Ute Schmid
<b>Sprache</b>	Deutsch/Englisch

---

<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<ul style="list-style-type: none"> <li>• Ghallab, Nau, Traverso: Automated Planning</li> <li>• Russell &amp; Norvig: Artificial Intelligence -- A Modern Approach</li> <li>• Wooldridge: An Introduction to Multiagent Systems</li> <li>• Schöning: Logik für Informatiker</li> <li>• Sterling, Shapiro: Prolog</li> </ul>
<b>Prüfungen</b>	Written Test Machine Learning

### ***Lehrveranstaltung Practice Intelligent Agents***

<b>Inhalte</b>	Practicing methods and techniques introduced in the lecture. Partly involving PROLOG programming.
<b>Dozenten</b>	Michael Siebers
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung

<b>Prüfungen</b>	Written Test Machine Learning
------------------	-------------------------------

### ***Prüfung Written Test Machine Learning***

<b>Beschreibung</b>	<p>Written Test Machine Learning (90 minutes)</p> <p>A total of 90 points can be reached in the exam. During the semester one- or two-week assignments are given. The provided solutions will be assessed and bonus points will be assigned. These points are included when calculating the final mark of the exam. At least 40 points (without bonus points) are required to pass the exam. It is possible to receive 1.0 without bonus points.</p> <p>Permitted utilities: lecture slides, additional material from lecture and practice, personal notes, calculator</p>
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

---

## Modul KTR-Datkomm-B: Datenkommunikation

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4P Pflichtbereich Grundlagen KuVS
<b>Lernziele / Kompetenzen</b>	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.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/ktr/leistungen/studium/">http://www.uni-bamberg.de/ktr/leistungen/studium/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	<ul style="list-style-type: none"><li>• erfolgreich abgeschlossene Prüfungen der Grundlagenfächer des Bachelorstudiums</li></ul>
<b>Inhaltlich</b>	Modul Einführung in die Informatik (DSG-EidI-B)
<b>vorausgesetzte Module</b>	Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (GdI-MfI-1) Modul Algorithmen und Datenstrukturen (MI-AuD-B)
<b>Bedingung für ECTS- Punkte</b>	Bestehen einer schriftlichen Prüfung in Form einer Klausur. 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>ECTS-Punkte</b>	6
<b>Bemerkung</b>	

### ***Lehrveranstaltung Vorlesung Datenkommunikation***

<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 Protokoll-Architekturen 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 Multiplexechniken 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 Paketverkehrlenkung dargestellt. Darüber hinaus werden die Grundfunktionen der Transportschicht und ihre exemplarische Umsetzung in TCP erläutert.

**Dozenten** Prof. Dr. Udo Krieger

**Sprache** Deutsch/Englisch

**Lehrformen** Vorlesung (V)

**Häufigkeit** WS, jährlich (jährlich )

**SWS** 2

**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, 2008
- Comer, D.: Computernetzwerke und Internets, Pearson Studium, München, 2004

Weitere Angaben und Erläuterungen erfolgen in der 1. Vorlesung.

**Prüfungen** Datenkommunikation

### ***Lehrveranstaltung Übung Datenkommunikation***

**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.

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. Die Berechnungs-, Vergabe- und Anrechnungsmodalitäten der Bonuspunktregelung werden zu Beginn der Lehrveranstaltung festgelegt und den Studierenden zur Kenntnis gebracht.

**Dozenten**

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

**Sprache**

Deutsch/Englisch

**Lehrformen**

Übung (Ü)

**Häufigkeit**

WS, jährlich (jährlich )

**SWS**

2

**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, 2008
- Comer, D.: Computernetzwerke und Internets, Pearson Studium, München, 2004

Weitere Literatur wird in der Übung benannt.

**Prüfungen**

Datenkommunikation

**Prüfung Datenkommunikation**

**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. Die Berechnungs-, Vergabe- und Anrechnungsmodalitäten der Bonuspunkteregelung werden zu Beginn der Lehrveranstaltung festgelegt und den Studierenden zur Kenntnis gebracht.

Zulässige Hilfsmittel der Prüfung:

- Taschenrechner ohne vollständige alphanumerische Tastatur und Grafikdisplay

**Typ** Klausur

**Dauer** 90 Minuten

## Modul KTR-GIK-M: Grundbausteine der Internet-Kommunikation

<b>Modulgruppen</b>	A4 Komplexe und Verteilte Systeme->A4P Pflichtbereich Grundlagen KuVS
<b>Lernziele / Kompetenzen</b>	<p>Wichtige Fertigkeiten zur Bewertung aktueller Kommunikationstechnologien sind nur durch die Vermittlung praktischer Fähigkeiten und Erfahrungen in team-orientierten Prozessen unter Zeit- und Zielvorgaben industrienah erlernbar. Die Studierenden werden in der Vorlesung Grundbausteine der Internet-Kommunikation und den begleitenden Laborübungen zu eigenverantwortlichem, team-orientierten Arbeiten angeleitet. Ziel ist der Erwerb praktischer Fertigkeiten auf dem Gebiet der IP-gestützten Datenkommunikation und die Fähigkeit, Lösungsvorschläge der modernen Internet-Kommunikation sicher beurteilen zu können.</p> <p>Die Lehrveranstaltung "Grundbausteine der Internet-Kommunikation" hat folgende Zielsetzungen:</p> <ul style="list-style-type: none"><li>• Fortführung der Vorlesung Datenkommunikation des Bachelorprogrammes als Profilbildungsstudium auf Masterniveau</li><li>• praktisches Erarbeiten der Grundlagen der Internet- und Multimedia-Kommunikation</li><li>• Aufbau und Verkehrsanalyse von TCP/IP-basierten Rechnernetzen mit modernen Echtzeit- und Web-Anwendungen</li><li>• Angebot einer Prüfungsalternative zur Lehrveranstaltung Multimedia-Kommunikation in Hochgeschwindigkeitsnetzen (KTR-MMK-M) oder Mobilkommunikation (KTR-Mobi-M) im Prüfungsfach Kommunikationssysteme und Rechnernetze</li><li>• Ergänzung der Lehrangebote in Verteilten Systemen und Medieninformatik zur Bildung eines Studienschwerpunktes "Mobile verteilte Systeme" bzw. Next Generation Systems</li></ul> <p>Die Lehrveranstaltung ist für Bachelor-Studierende im Profilbildungsstudium zur Stärkung ihrer Arbeitsmarktchancen sowie für Austauschstudenten/innen besonders empfehlenswert.</p>
<b>WWW</b>	<a href="http://www.uni-bamberg.de/ktr/leistungen/studium/">http://www.uni-bamberg.de/ktr/leistungen/studium/</a>
<b>Arbeitsaufwand:</b>	180 Stunden

---

<b>Empfohlene Vorkenntnisse</b>	<ul style="list-style-type: none"> <li>• Datenkommunikation im Umfang KTR-Datkomm-B</li> <li>• Programmierkenntnisse in JAVA (oder C++)</li> <li>• der Erwerb von LINUX-Kenntnissen wird empfohlen, ist aber keine Voraussetzung</li> </ul>
<b>Inhaltlich vorausgesetzte Module</b>	<p>Modul &lt;Keine Modulbindung&gt; ()</p> <p>Modul Einführung in die Informatik (DSG-EidI-B)</p> <p>Modul Datenkommunikation (KTR-Datkomm-B)</p> <p>Modul Algorithmen und Datenstrukturen (MI-AuD-B)</p>
<b>Bedingung für ECTS-Punkte</b>	<p>Es werden die Leistungen der als Gruppenarbeit ausgeführten schriftlichen Ausarbeitung der Aufgabenstellungen und ihrer Präsentation sowie die Ergebnisse einer individuellen mündlichen Kolloquiumsprüfung bewertet. Der Arbeitsaufwand gliedert sich grob wie folgt:</p> <ul style="list-style-type: none"> <li>• Teilnahme an Präsenzveranstaltungen (Vorlesung, Laborübungen, Laborbesprechungen): 45 Stunden</li> <li>• Vorbereitung, Ausführung und Nachbereitung von Vorlesungen und Laborübungen: 100 Stunden</li> <li>• Prüfungsvorbereitung: 35 Stunden</li> </ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	The module can be selected by exchange students and master students speaking only English.

### ***Lehrveranstaltung Grundbausteine der Internet-Kommunikation***

<b>Inhalte</b>	<p>Die Lehrveranstaltung vermittelt eine Einführung in die theoretischen Grundlagen wichtiger kommunikationstechnischer Problemstellungen zu den Themengebieten Grundlagen der Internet-Kommunikation, Verbindungssegmente und Routing in IP-Netzen, Transportprotokolle in IP-Netzen bzw. fortgeschrittener Module wie Echtzeit-Kommunikation und Sicherheit in IP-Netzen und die eigenständige praktische Umsetzung des erworbenen Wissens durch vorgegebene Laborübungen zur Internet-Kommunikation in Kleingruppen. Dabei werden weitere Hilfsmittel und Anleitungen sowie die Laborumgebung bereitgestellt.</p> <p>Zur Implementierung soll ein Rechnernetz im Labor konfiguriert und getestet werden. Die Betriebssystem-Grundausrüstung und erforderliche Software-Werkzeuge wie Wireshark und Atheris werden bereitgestellt. Grundlagen der Handhabung werden von den Studierenden im Projekt selbst erarbeitet.</p>
----------------	--

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:

- Unterteilung der Arbeiten in Arbeitspakete (laboratories/work packages),
- ihre Untergliederung in Aufgaben (tasks) und Teilaufgaben (subtasks) mit Meilensteinen
- und der Darlegung von Zwischenergebnissen bzw.
- einem Abschlussbericht mit Abschlusspräsentation

Weitere Laboraufgaben zu aktuellen Forschungsfragen im "Future Generation Internet" werden bei Bedarf in die Lehrveranstaltung integriert. Details werden in der Vorlesung angekündigt.

Eine aktuelle Liste der bearbeiteten Themen der Lehrveranstaltung wird in der Vorlesung bereitgestellt.

**Dozenten**

Prof. Dr. Udo Krieger  
Philipp Eittenberger

**Sprache**

Englisch/Deutsch

**Lehrformen**

Vorlesung und Übung (V/Ü)

**Häufigkeit**

jährlich nach Bedarf WS und SS

**SWS**

4

**Literatur**

Grundlagen:

- J. Liebeherr, M. Elzarki: Mastering Networks, An Internet Lab Manual, Pearson Education, Boston, 2004.

weitere Literatur zu einzelnen Arbeitspaketen:

- Kurose, J., Ross, K.W.: Computernetzwerke – ein Top-Down-Ansatz mit Schwerpunkt Internet, Pearson Studium, München, 2008 .
- Tanenbaum, A. S.: Computernetzwerke, Pearson Studium, München, 4. Aufl., 2003.
- Sikora, A.: Technische Grundlagen der Rechnerkommunikation, Fachbuchverlag Leipzig, 2003.
- Leon-Garcia, A., Widjaja, I.: Communication Networks, McGraw-Hill, Boston, 2nd ed. 2004.
- Badach, A.: Voice over IP - Die Technik, Carl Hanser Verlag, München, 2. Aufl., 2005.

- Flaig, G., u.a.: Internet-Telefonie, Open source Press, München, 2006.

Eine aktualisierte Liste wird in der Vorlesung bereitgestellt.

**Prüfungen**

Grundbausteine der Internet-Kommunikation

***Prüfung Grundbausteine der Internet-Kommunikation*****Beschreibung**

Die Leistungsbewertung der Lehrveranstaltung erfolgt nach Abschluss auf folgender Grundlage:

- Auswertung des in Gruppenarbeit gemeinsam erstellten schriftlichen Projektberichtes der bearbeiteten Aufgaben und der Abschlusspräsentation der Projektgruppen (40% der Endbewertung)
- Vorführung und Erläuterungen der Zusammenhänge einzelner Aufgaben und Ergebnisse im Rahmen einer individuellen Kolloquiumsprüfung im Umfang von 20 Minuten (60% der Endbewertung)

Die individuelle Gesamtleistung muss mit der Note "ausreichend" bewertet werden, um die Prüfung zu bestehen.

**Typ**

Hausarbeit, Kolloquium

**Dauer**

20 Minuten

## Modul KTR-Mfl-2: Mathematik für Informatiker 2 (Lineare Algebra)

<b>Modulgruppen</b>	A1 Mathematische Grundlagen
<b>Lernziele / Kompetenzen</b>	<p>Die mathematischen Grundlagen der Informatik aus dem Bereich der linearen Algebra werden den Studierenden vermittelt. Der besondere Bezug zur Angewandten Informatik bzw. Software Systems Science wird in den Vorlesungsbeispielen und Übungen herausgearbeitet.</p> <p>Die Studierenden lernen, die grundlegende Methoden und Algorithmen anzuwenden und spezifische Anwendungen der Angewandten Informatik als Probleme der linearen Algebra zu erkennen, zu formulieren und mit Hilfe geeigneter Verfahren zu lösen.</p>
<b>WWW</b>	<a href="http://www.uni-bamberg.de/ktr/leistungen/studium/">http://www.uni-bamberg.de/ktr/leistungen/studium/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	-
<b>Inhaltlich vorausgesetzte Module</b>	Modul Mathematik-Vorkurs-Bachelorstudium (KTR-MVK-B)
<b>Bedingung für ECTS- Punkte</b>	<p>Bestehen einer schriftlichen Prüfung im Umfang von 90 Minuten. 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>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	Das Modul stellt die Grundlagen für Studierende der Angewandten Informatik und Software Systems Science sowie Studierende im Nebenfach verwandter Bachelor-Studiengänge der Fakultät WIAI bereit.

### ***Lehrveranstaltung Mathematik für Informatiker 2 (Lineare Algebra)***

<b>Inhalte</b>	<p>Die Lehrveranstaltung stellt mathematische Grundlagen der Informatik bereit und ist dem Pflichtbereich der Modulgruppe A1 "Mathematische Grundlagen" für Angewandte Informatik/Software Systems Science zugeordnet.</p> <p>Es werden grundlegende Methoden und Algorithmen der Gruppen- und Ringtheorie, der linearen Algebra, der Matrizenalgebra, der</p>
----------------	--



Lösungstheorie linearer Gleichungssysteme, der Eigenwerttheorie sowie spezifische Anwendungen der Informatik vorgestellt.

<b>Dozenten</b>	Prof. Dr. Udo Krieger
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung und Übung (V/Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	4
<b>Literatur</b>	<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, Vieweg+Teubner, Bachelorkurs Mathematik, 2011.</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>
<b>Prüfungen</b>	KTR-Mathematik-Informatik-2
<b><i>Prüfung KTR-Mathematik-Informatik-2</i></b>	
<b>Beschreibung</b>	schriftliche Prüfung zu Inhalten der Vorlesung und Übungen im Umfang von 90 Minuten
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## Modul KTR-Sem-B: Bachelorseminar

<b>Modulgruppen</b>	A7 Seminare und Projekte->Seminare
<b>Lernziele / Kompetenzen</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>WWW</b>	<a href="http://www.uni-bamberg.de/ktr/leistungen/studium/">http://www.uni-bamberg.de/ktr/leistungen/studium/</a>
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Module gemäß der folgenden Spezifikation
<b>Inhaltlich vorausgesetzte Module</b>	Modul Datenkommunikation (KTR-Datkomm-B) Modul Algorithmen und Datenstrukturen (MI-AuD-B)
<b>Bedingung für ECTS- Punkte</b>	Eine als ausreichend bewertete schriftliche Ausarbeitung des Seminarthemas mit einem erfolgreich vorgestellten Referat sind zum Bestehen der Modulprüfung erforderlich.  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>ECTS-Punkte</b>	3

### ***Lehrveranstaltung Bachelorseminar KTR***

<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 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, logisch korrekter, verständlicher Form.
<b>Dozenten</b>	Prof. Dr. Udo Krieger Philipp Eittenberger
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Seminar (S)

<b>Häufigkeit</b>	jährlich nach Bedarf WS und SS
<b>SWS</b>	2
<b>Literatur</b>	Die aktuelle Literaturliste wird bei der Vorbesprechung bereitgestellt.
<b>Prüfungen</b>	Seminar KTR-Bachelor

### ***Prüfung Seminar KTR-Bachelor***

<b>Beschreibung</b>	Die Gesamtnote ergibt sich zu gleichen Teilen aus der Bewertung der schriftliche Ausarbeitung und des Referates und muss mit mindestens ausreichend bewertet sein.
<b>Typ</b>	Hausarbeit, Referat
<b>Dauer</b>	30 Minuten

## **Modul Mathe-B-01: Mathematik für Wirtschaftswissenschaftler I (Analysis)**

**Modulgruppen** A1 Mathematische Grundlagen

**Lernziele /  
Kompetenzen** Vermittlung von mathematischen Grundkenntnissen aus dem Gebiet der Analysis. Die Teilnehmer dieser Vorlesung/Übung sollen in die Lage versetzt werden, die mathematischen Verfahren und Konzepte der weiterführenden (wirtschafts-)informatischen Veranstaltungen zu verstehen und zu beherrschen.

**WWW** [http://www.uni-bamberg.de/sowi/fachgebiete/weitere\\_faecher/wirtschaftsmathematik/](http://www.uni-bamberg.de/sowi/fachgebiete/weitere_faecher/wirtschaftsmathematik/)

**Arbeitsaufwand:** 90 Stunden

**Empfohlene  
Vorkenntnisse**

**Inhaltlich** -  
**vorausgesetzte Module**

**Bedingung für ECTS-  
Punkte** Bestehen der Prüfung

**ECTS-Punkte** 3

**Bemerkung**

### ***Lehrveranstaltung Vorlesung/Übung Mathematik für Wirtschaftswissenschaftler I***

**Inhalte**

- 0 Grundlagen
- 0.1 Kartesische Produkte und Relationen
- 0.2 Abbildungen
- 1 Folgen und Reihen
  - 1.1 Folgen
  - 1.2 Reihen
  - 1.3 Finanzmathematik
    - 1.3.1 Einfache Zinsrechnung
    - 1.3.2 Zinseszinsrechnung
    - 1.3.3 Rentenrechnung
    - 1.3.4 Tilgungsrechnung
    - 1.3.5 Investitionsrechnung
  - 2 Differenzialrechnung
    - 2.1 Funktionen einer und mehrerer Variablen

- 
- 2.1.1 Beispiele, grafische Darstellung und Eigenschaften von Funktionen einer und mehrerer Variablen
  - 2.1.2 Polynome, gebrochen rationale und algebraische Funktionen
  - 2.1.3 Transzendente Funktionen (Exponential-, Logarithmus- und Winkelfunktionen)
  - 2.1.4 Grenzwerte und Stetigkeit von Funktionen
  - 2.2 Differenzialrechnung für Funktionen einer Variablen
    - 2.2.1 Differenzialquotient und Ableitungsregeln
    - 2.2.2 Differenziation der Grundfunktionen
    - 2.2.3 Monotonie, Konvexität/Konkavität und Extremstellen differenzierbarer Funktionen einer Variablen
    - 2.2.4 Rechnen mit dem Symbol #, die Regeln von de l'Hospital
    - 2.2.5 Approximation differenzierbarer Funktionen durch Polynome, Differenziale und der Satz von Taylor
    - 2.2.6 Elastizitäten
  - 2.3 Differenzialrechnung für Funktionen mehrerer Variablen
    - 2.3.1 Partielle und totale Ableitungen
    - 2.3.2 Die Kettenregel für Funktionen mehrerer Variablen
    - 2.3.3 Partielle Ableitungen höherer Ordnung
    - 2.3.4 Partielle und totale Differenziale, partielle Elastizitäten
    - 2.3.5 Implizite Funktionen
    - 2.3.6 Extremstellen differenzierbarer Funktionen mehrerer Variablen (ohne Nebenbedingungen)
    - 2.3.7 Extremstellen differenzierbarer Funktionen mehrerer Variablen (mit Nebenbedingungen)
    - 2.3.8 Differenziation vektorwertiger Funktionen
  - 3 Integralrechnung
    - 3.1 Das unbestimmte Integrale
    - 3.2 Das bestimmte Integrale
    - 3.3 Uneigentliche Integrale
    - 3.4 Integralrechnung für Funktionen mehrerer Variablen

**Dozenten** Dr. rer. pol. Reinhard Dobbener

**Sprache** Deutsch

**Lehrformen** Vorlesung und Übung (V/Ü)

**Häufigkeit** WS, SS (jedes Semester)

**SWS** 3

**Literatur**

**Prüfungen**                      Mathematik für Wirtschaftswissenschaftler I

***Prüfung Mathematik für Wirtschaftswissenschaftler I***

**Beschreibung**

**Typ**                                      Klausur

**Dauer**                                    60 Minuten

# Modul MI-AuD-B: Algorithmen und Datenstrukturen

<b>Modulgruppen</b>	A2 Allgemeine Informatik Grundlagen
<b>Lernziele / Kompetenzen</b>	Das Modul vermittelt die Fähigkeit, die Qualität von Datenstrukturen und Algorithmen im Hinblick auf konkrete Anforderungen einzuschätzen und ihre Implementierung in einem Programm umzusetzen. Daneben sollen grundlegende Kenntnisse im Bereich der Algorithmenkonstruktion erworben werden. Durch die Übung soll auch Sicherheit im Umgang mit objektorientierten Entwicklungsmethoden und Standardbibliotheken erworben und Teamarbeit geübt werden.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/?id=18671">http://www.uni-bamberg.de/?id=18671</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in Informatik und Programmierung, wie sie z. B. in der Veranstaltung DSG-EiAPS-B im Rahmen des Moduls DSG-EidI-B vermittelt werden.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur
<b>ECTS-Punkte</b>	6
<b>Bemerkung</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 6 Teilleistungen: insgesamt ca. 60 Stunden (inkl. 15 Stunden [= 2/3] der 2 SWS Übungsbetrieb)</li> <li>• Klausurvorbereitung: ca. 37,5 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li> </ul>

## ***Lehrveranstaltung Vorlesung Algorithmen und Datenstrukturen***

<b>Inhalte</b>	Die Vorlesung betrachtet die klassischen Bereiche des Themengebiets Algorithmen und Datenstrukturen: <ul style="list-style-type: none"> <li>• Einleitung</li> </ul>
----------------	---

- Listen
- Hashverfahren
- Bäume
- Graphen
- Sortieren
- Algorithmenkonstruktion

<b>Dozenten</b>	Prof. Dr. Andreas Henrich
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Eines der Standardlehrbücher über Algorithmen und Datenstrukturen, z.B.: <ul style="list-style-type: none"><li>• Saake, Gunter; Sattler, Kai-Uwe: Algorithmen und Datenstrukturen: Eine Einführung mit Java, ISBN: 978-3-89864-385-6, 3. Aufl. 2006, 512 Seiten, Dpunkt Verlag</li><li>• Ottmann, Thomas; Widmayer, Peter: Algorithmen und Datenstrukturen, ISBN: 978-3-8274-1029-0, 4. Aufl. 2002, 736 Seiten, Spektrum, Akaedemischer Verlag</li></ul>
<b>Prüfungen</b>	Algorithmen und Datenstrukturen (Klausur)

### ***Lehrveranstaltung Übung Algorithmen und Datenstrukturen***

<b>Inhalte</b>	In der Übung werden folgende Aspekte betrachtet: <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>• Nutzung von Bibliotheken</li><li>• Anwendung von Prinzipien zur Algorithmenkonstruktion</li></ul>
----------------	---

<b>Dozenten</b>	Mitarbeiter Medieninformatik
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich (jährlich im Sommersemester)
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung
<b>Prüfungen</b>	Algorithmen und Datenstrukturen (Klausur)



**Prüfung Algorithmen und Datenstrukturen (Klausur)**

<b>Beschreibung</b>	<p>In der <b>Klausur</b> können 90 Punkte erzielt werden.</p> <p>Im Semester werden darüber hinaus 6 <b>Teilleistungen</b> zur Bearbeitung ausgegeben. Für jede Teilleistung stehen 2 Wochen als Bearbeitungszeit zur Verfügung. Die Lösungen zu den Teilleistungen werden bewertet. Pro Teilleistung können maximal 2 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 (also maximal 12 Punkte) als Bonuspunkte angerechnet. Eine 1,0 ist dabei aber auf jeden Fall auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar.</p>
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## **Modul MI-LA-DatSchu-B: Grundlagen und Fallstudien zum Datenschutz**

<b>Modulgruppen</b>	A6 Kontextstudium->Allgemeine Schlüsselqualifikationen
<b>Lernziele / Kompetenzen</b>	Vermittlung der erforderlichen fachlichen Kenntnisse, Methoden und Fähigkeiten, um die inhaltlichen, organisatorischen und technischen Anforderungen des Datenschutzes und der Datensicherheit in einem Unternehmen umsetzen zu können.  Kenntnis der Grundprinzipien des Datenschutzes und der Datensicherheit, der gesetzlichen Anforderungen und der datenschutzrelevanten Rechtsprechung.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/?id=3896">http://www.uni-bamberg.de/?id=3896</a>
<b>Arbeitsaufwand:</b>	120 Stunden
<b>Empfohlene Vorkenntnisse</b>	-
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur
<b>ECTS-Punkte</b>	4
<b>Bemerkung</b>	Der typische Aufwand zum Modul gliedert sich grob wie folgt: <ul style="list-style-type: none"><li>• 45 Stunden für den Besuch der Veranstaltung</li><li>• 60 Stunden für die Nachbereitung und die Betrachtung von Fallstudien</li><li>• 15 Stunden Prüfungsvorbereitung und Prüfung</li></ul>

### ***Lehrveranstaltung Grundlagen und Fallstudien zum Datenschutz***

<b>Inhalte</b>	Gliederung der Veranstaltung <ol style="list-style-type: none"><li>1. Ziel des Datenschutzes</li><li>2. Grundlagen des BDSG</li><li>3. Allgemeine Vorschriften des BDSG</li><li>4. Datenschutz im nicht-öffentlichen Bereich</li></ol>
<b>Dozenten</b>	Dr. theol. M.A. phil. Wolfgang Hübner
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung und Übung (V/Ü)

<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	4
<b>Literatur</b>	wird in der Veranstaltung bekannt gegeben
<b>Prüfungen</b>	Datenschutz (Klausur)
<b><i>Prüfung Datenschutz (Klausur)</i></b>	
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## Modul MI-WebT-B: Web-Technologien

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3W-II Wahlpflichtbereich II Schwerpunkt PuSWE
<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 selbständig mit gängigen Frameworks und Techniken zu entwickeln.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/?id=6437">http://www.uni-bamberg.de/?id=6437</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundkenntnisse der Informatik und zu Dateiformaten, wie Sie z.B. in den unten angegebenen Modulen erworben werden können. Insbesondere sind auch Kenntnisse in einer imperativen oder objektorientierten Programmiersprache erforderlich.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Einführung in die Informatik (DSG-EidI-B) Modul Einführung in die Medieninformatik (MI-EMI-B)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur
<b>ECTS-Punkte</b>	6
<b>Bemerkung</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>• Klausurvorbereitung: ca. 37,5 Stunden (basierend auf dem bereits im obigen Sinne erarbeiteten Stoff)</li></ul>

### **Lehrveranstaltung Vorlesung Web-Technologien**

---

<b>Inhalte</b>	Die Veranstaltung betrachtet die Aufgabenfelder, Konzepte und Technologien zur Entwicklung von Web-Anwendungen. Folgende Bereiche bilden dabei die Schwerpunkte der Veranstaltung: <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: die Basics &amp; AJAX</li> <li>• Server-Side Scripting: CGI + PHP</li> <li>• Frameworks: CakePHP, RubyOnRails</li> <li>• Sicherheit von Web-Anwendungen</li> <li>• CMS, LMS, SEO &amp; Co.</li> </ul>
----------------	--

<b>Dozenten</b>	Prof. Dr. Andreas Henrich
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	aktuelle Literatur wird in der Veranstaltung bekannt gegeben
<b>Prüfungen</b>	Web-Technologien (Klausur)

### ***Lehrveranstaltung Übung Web-Technologien***

<b>Inhalte</b>	praktische Aufgaben zum Stoff der Vorlesung
<b>Dozenten</b>	Mitarbeiter Medieninformatik
<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	siehe Vorlesung
<b>Prüfungen</b>	Web-Technologien (Klausur)

### ***Prüfung Web-Technologien (Klausur)***

<b>Beschreibung</b>	In der <b>Klausur</b> können 90 Punkte erzielt werden.  Im Semester werden darüber hinaus 3 <b>Teilleistungen</b> zur Bearbeitung ausgegeben. Für jede Teilleistung stehen in der Regel 4 Wochen als Bearbeitungszeit zur Verfügung. Die 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 (also maximal 12 Punkte) als Bonuspunkte angerechnet. Eine 1,0 ist dabei aber auf jeden Fall auch ohne Punkte aus der Bearbeitung der Teilleistungen erreichbar.

**Typ** Klausur

**Dauer** 90 Minuten

## Modul SEDA-DMS-B: Datenmanagementsysteme

<b>Modulgruppen</b>	A2 Allgemeine Informatik Grundlagen
<b>Lernziele / Kompetenzen</b>	Das Modul vermittelt eine systematische Einführung in das Gebiet der Datenmanagementsysteme. 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 mit ERM und SERM. Die Studierenden erlernen die Grundlagen der Sprache SQL und können mit SQL Datenbankschemata generieren sowie zugehörige Datenbanken aufbauen und manipulieren. Schließlich sammeln sie erste Erfahrungen im Umgang mit realen Datenbankverwaltungssystemen.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/fakultaeten/wiai/faecher/wirtschaftsinformatik/seda/leistungen/studium/">http://www.uni-bamberg.de/fakultaeten/wiai/faecher/wirtschaftsinformatik/seda/leistungen/studium/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Kenntnisse aus dem Modul Grundlagen betrieblicher Informationssysteme sind wünschenswert, jedoch nicht Voraussetzung
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Abschlussklausur. Der Arbeitsaufwand von 180 Stunden gliedert sich in etwa wie folgt: <ul style="list-style-type: none"> <li>• 60 Stunden Teilnahme an Vorlesung und Übung</li> <li>• 120 Stunden Selbststudium</li> </ul>
<b>ECTS-Punkte</b>	6

### ***Lehrveranstaltung Vorlesung Datenmanagementsysteme***

<b>Inhalte</b>	Datenmanagementsysteme sind zentrale Teilsysteme betrieblicher Anwendungssysteme. Ihre Entwicklung und ihr Betrieb stellen Kernaufgaben der Wirtschaftsinformatik dar. Das Modul vermittelt eine systematische Einführung in diesen Themenbereich. Der Fokus liegt dabei auf der Analyse, der Gestaltung und der Nutzung von Datenmanagementsystemen, nicht etwa auf der Implementierung von Datenbankverwaltungssystemen.  Inhaltliche Schwerpunkte bilden das Relationenmodell, die Sprache SQL, Architekturen von Datenmanagementsystemen, der Entwurf von
----------------	---

Datenbankschemata, theoretische Grundlagen der Datenmodellierung, Transaktionen und Transaktionsverwaltung sowie der Betrieb von Datenmanagementsystemen.

Praktische Fertigkeiten werden insbesondere in Bezug auf den Entwurf von Datenbankschemata und SQL vermittelt. SQL wird anhand von konkreten Datenbankverwaltungssystemen geübt. Fertigkeiten werden insbesondere in Bezug auf SQL vermittelt.

Inhalte:

- Einführung
- Das Relationenmodell
- Die Sprache SQL
- Architekturen von Datenmanagementsystemen
- Entwurf von Datenbankschemata
- Fallstudie: Entwicklung eines Datenmanagementsystems
- Theoretische Grundlagen der Datenmodellierung
- Transaktionen und Transaktionsverwaltung
- Betrieb von datenbankbasierten AWS
- Alternative Entwicklungen im Bereich Datenbanken

<b>Dozenten</b>	Prof. Dr. Elmar J. Sinz
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<ul style="list-style-type: none"><li>• Date C.J.: An Introduction to database systems. 8th Edition, Addison-Wesley, Reading, Massachusetts 2003</li><li>• Ferstl O.K., Sinz E.J.: Grundlagen der Wirtschaftsinformatik. 6. Auflage, Oldenbourg, München 2008, Kapitel 9.2</li><li>• Kemper A., Eickler A.: Datenbanksysteme. Eine Einführung. 8. Auflage, Oldenbourg, München 2011</li><li>• Pernul G., Unland R.: Datenbanken im Unternehmen. Analyse, Modellbildung und Einsatz. 2. Auflage, Oldenbourg, München 2003</li><li>• Coronel C., Morris S., Rob P.: Database Systems. Design, Implementation, and Management. 9th Edition, Course Technology, Thomson Learning, Boston 2009</li></ul>



- Vossen G.: Datenbankmodelle, Datenbanksprachen und Datenbankmanagement-Systeme. 5. Auflage, Oldenbourg, München 2008

**Prüfungen** Datenmanagementsysteme

### ***Lehrveranstaltung Übung Datenmanagementsysteme***

**Inhalte** Die Inhalte der Vorlesung werden anhand von Übungsaufgaben und Fallbeispielen vertieft. Praktische Übungen werden unter Verwendung eines gängigen Datenbankverwaltungssystems durchgeführt.

**Dozenten** Mitarbeiter Wirtschaftsinformatik, insb. Systementwicklung und Datenbankanwendung

**Sprache** Deutsch

**Lehrformen** Übung (Ü)

**Häufigkeit** SS, jährlich

**SWS** 2

**Literatur** siehe Vorlesung

**Prüfungen** Datenmanagementsysteme

### ***Prüfung Datenmanagementsysteme***

**Typ** Klausur

**Dauer** 90 Minuten

## **Modul SSS-AIS-B: Bachelorseminar aus der Angewandten Informatik**

**Modulgruppen** A7 Seminare und Projekte->Seminare

**Lernziele /**

**Kompetenzen**

**WWW** -

**Arbeitsaufwand:** 90 Stunden

**Empfohlene**

**Vorkenntnisse**

**Inhaltlich** -

**vorausgesetzte Module**

**Bedingung für ECTS-Punkte** Näheres regeln die Modulbeschreibungen zu den Bachelorseminaren der Lehreinheiten in Angewandter Informatik (siehe Modulhandbuch zum Studiengang B.Sc. Angewandte Informatik).

**ECTS-Punkte** 3

**Bemerkung**

### ***Lehrveranstaltung Bachelorseminar aus der Angewandten Informatik***

**Inhalte**

**Dozenten** -

**Sprache** Deutsch/Englisch

**Lehrformen** Seminar (S)

**Häufigkeit** WS, SS

**SWS** 2

**Literatur**

**Prüfungen** -

# Modul SSS-BAP-B: Bachelorprojekt Software Systems Science

<b>Modulgruppen</b>	A7 Seminare und Projekte->Projekt
<b>Lernziele / Kompetenzen</b>	Studierende sollen ein vertieftes Verständnis der bei der Durchführung von Softwaresystem-Projekten auftretenden konzeptionellen und praktischen Probleme wie auch von erfolgsversprechenden Lösungsansätzen zu diesen Problemen erhalten. Da dies anhand der intensiven Bearbeitung eines Themas aus dem Forschungsbereich von Softwaresystemen in Kleingruppen - oder ggf. auch 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 Präsentation dieser Ergebnisse.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	360 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende methodische Kenntnisse zur Planung und Durchführung von Softwareprojekten, z.B. Erworben im Modul "Software Engineering Lab" (SWT-SWL-B)
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der beiden schriftlichen Hausarbeiten (Zwischen- und Abschlussbericht) und der Präsentation des Projektposters (Referat); aktive Teilnahme am Planungsworkshop und an den Projekttreffen.  Die Modulnote ergibt sich aus folgender Gewichtung der Prüfungsleistungen: 25% Zwischenbericht; 50% Abschlussbericht; 25% Posterpräsentation. Berichte und Posterpräsentation dürfen wahlweise in Deutsch oder Englisch abgefasst sein.  Arbeitsaufwand 360 Std., welche sich grob wie folgt gliedern: <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. Abfassung des Zwischenberichts</li> <li>• 40 Std. Abfassung des Abschlussberichts</li> </ul>

- 20 Std. Erstellung und Präsentation des Projektposters

**ECTS-Punkte** 12

**Bemerkung**

## ***Lehrveranstaltung Projektübung zu Bachelorprojekt Software Systems Science***

**Inhalte**

Ein überschaubares Projekt zu Themen aus der aktuellen Forschungsarbeit auf den Gebieten

- Theoretische Grundlagen von Softwaresystemen
- Kommunikationsdienste und vernetzte Softwaresysteme
- Komplexe und verteilte Softwaresysteme
- Programmierung und Entwicklung 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, deren Ergebnisse in einem Planungsworkshop vorgestellt und diskutiert werden.

Begleitend zur Durchführung des Projekts 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. Der Fortschritt bzw. die Ergebnisse des Projekts werden von Studierenden in einem Zwischen- bzw. 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 und erläutert wird.

**Dozenten**

Mitarbeiter Praktische Informatik

Mitarbeiter Informatik, insbesondere Kommunikationsdienste, Telekommunikationssysteme und Rechnernetze

Prof. Dr. Gerald Lüttgen

Michael Mendler

Prof. Dr. Udo Krieger

---

Prof. Dr. Guido Wirtz  
Mitarbeiter Grundlagen der Informatik  
Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und  
Programmiersprachen

<b>Sprache</b>	Deutsch/Englisch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, SS
<b>SWS</b>	8
<b>Literatur</b>	Je nach Projektthematik; wird zu Beginn eines jeden Semesters bekannt gegeben.
<b>Prüfungen</b>	Bachelorprojekt Software Systems Science (Zwischenbericht) Bachelorprojekt Software Systems Science (Posterpräsentation) Bachelorprojekt Software Systems Science (Abschlussbericht)

### ***Prüfung Bachelorprojekt Software Systems Science (Zwischenbericht)***

<b>Beschreibung</b>	Anfertigen eines schriftlichen Zwischenberichts zum Projekt etwa nach 80 Std. geleisteter Projektarbeit, spätestens am Ende des Semesters, in dem das Projekt begonnen wurde.
<b>Typ</b>	Hausarbeit
<b>Dauer</b>	2 Wochen

### ***Prüfung Bachelorprojekt Software Systems Science (Posterpräsentation)***

<b>Beschreibung</b>	Referat auf der Grundlage des von der/dem Vortragenden im Projekt bearbeiteten Themas im Rahmen einer Abschlussveranstaltung, auf der das zum Projekt angefertigte Poster präsentiert und erläutert wird; hier können auch praktische Projektergebnisse (z.B. lauffähige Software) demonstriert werden.
<b>Typ</b>	Referat
<b>Dauer</b>	30 Minuten

### ***Prüfung Bachelorprojekt Software Systems Science (Abschlussbericht)***

<b>Beschreibung</b>	Anfertigen eines schriftlichen Abschlussberichts nach abgeschlossener Projektarbeit.
<b>Typ</b>	Hausarbeit
<b>Dauer</b>	4 Wochen

## Modul Stat-B-01: Methoden der Statistik I

<b>Modulgruppen</b>	A1 Mathematische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen mit den grundlegenden statistischen Methoden vertraut gemacht werden. Besondere Schwerpunkte bilden dabei die theoretischen Grundlagen dieser Methoden, die Voraussetzungen ihrer Anwendbarkeit, ihre Umsetzung in Statistiksoftware sowie die sinnvolle Interpretation der Ergebnisse.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/stat-oe/">http://www.uni-bamberg.de/stat-oe/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	-
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Prüfung
<b>ECTS-Punkte</b>	6

### ***Lehrveranstaltung Vorlesung Methoden der Statistik I***

<b>Inhalte</b>	Im Rahmen der Vorlesung Methoden der Statistik I und der zugehörigen Übung werden die wichtigsten Grundlagen und Methoden der deskriptiven (beschreibenden) Statistik vermittelt. Dabei umfasst der Abschnitt zur deskriptiven Statistik Methoden, mit denen ein gegebenes Datenmaterial überschaubar dargestellt bzw. durch wenige aussagekräftige Zahlen wie Lageparameter, Streuungsmaße oder Korrelationskoeffizienten charakterisiert werden kann. Schließlich werden verschiedene Fragen der Datenerhebung angesprochen, denn eine noch so ausgefeilte statistische Methode ist nur so gut, wie die Daten, auf die sie angewendet wird.
<b>Dozenten</b>	Prof. Dr. Susanne Rässler
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, SS
<b>SWS</b>	3
<b>Literatur</b>	-
<b>Prüfungen</b>	Methoden der Statistik I

***Lehrveranstaltung Übung Methoden der Statistik I***

<b>Inhalte</b>	-
<b>Dozenten</b>	Prof. Dr. Susanne Rässler
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, SS
<b>SWS</b>	1
<b>Literatur</b>	-
<b>Prüfungen</b>	Methoden der Statistik I

***Prüfung Methoden der Statistik I***

<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## Modul Stat-B-02: Methoden der Statistik II

<b>Modulgruppen</b>	A1 Mathematische Grundlagen
<b>Lernziele / Kompetenzen</b>	Die Studierenden sollen mit den grundlegenden statistischen Methoden vertraut gemacht werden. Besondere Schwerpunkte bilden dabei die theoretischen Grundlagen dieser Methoden, die Voraussetzungen ihrer Anwendbarkeit, ihre Umsetzung in Statistiksoftware sowie die sinnvolle Interpretation der Ergebnisse.
<b>WWW</b>	<a href="http://www.uni-bamberg.de/stat-oe/">http://www.uni-bamberg.de/stat-oe/</a>
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	-
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Prüfung
<b>ECTS-Punkte</b>	6

### ***Lehrveranstaltung Vorlesung Methoden der Statistik II***

<b>Inhalte</b>	<p>Im Rahmen der Vorlesung Methoden der Statistik II und der zugehörigen Übung werden die wichtigsten Grundlagen der Wahrscheinlichkeitsrechnung und der induktiven (schließenden) Statistik vermittelt. Im Einzelnen befasst sich die Vorlesung Methoden der Statistik II mit den grundlegenden Begriffen, Regeln und Gesetzmäßigkeiten der Wahrscheinlichkeitsrechnung, 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. Des Weiteren stehen in der Vorlesung zur induktiven Statistik 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</p>
----------------	---



---

Hypothesentests behandelt. Im Anschluss daran folgt ein Überblick über einige weitere interessante Teilgebiete der Statistik, wobei speziell die Methoden der Regressionsrechnung und der Analyse kategorialer Variablen ausführlicher besprochen werden.

<b>Dozenten</b>	Prof. Dr. Susanne Rässler
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, SS
<b>SWS</b>	3
<b>Literatur</b>	-
<b>Prüfungen</b>	Methoden der Statistik II

### ***Lehrveranstaltung Übung Methoden der Statistik II***

<b>Inhalte</b>	-
<b>Dozenten</b>	Prof. Dr. Susanne Rässler
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, SS
<b>SWS</b>	1
<b>Literatur</b>	-
<b>Prüfungen</b>	Methoden der Statistik II

### ***Prüfung Methoden der Statistik II***

<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## Modul SWT-CCP-M: Compiler Construction Project

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3W-II Wahlpflichtbereich II Schwerpunkt PuSWE
<b>Lernziele / Kompetenzen</b>	Students will apply and extend their knowledge gained in the module Principles of Compiler Construction (SWT-PCC-M) by writing a compiler from scratch in the functional language Haskell. This will cement the concepts of compiler construction by experiencing the process of translating theoretical ideas, using efficient algorithms and data structures, into arguably the most indispensable tool for Computer Scientists.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in Algorithmen und Datenstrukturen. Darüber hinaus sind Grundkenntnisse in einer funktionalen Programmiersprache, z.B. Erworben durch die Teilnahme an Teilen des Moduls GdI-NPP-B "Nicht-Prozedurale Programmierung", hilfreich.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Principles of Compiler Construction (SWT-PCC-M)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung Arbeitsaufwand 180 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"><li>• 60 Std. Teilnahme an den Übungen: Entwicklung und Programmierung von Compiler-Komponenten, und Besprechung der produzierten Softwarekomponenten mit dem Betreuer</li><li>• 100 Std. Bearbeitung der schriftlichen Hausarbeit</li><li>• 20 Std. Vorbereitung auf das Kolloquium</li></ul>
<b>ECTS-Punkte</b>	6

### ***Lehrveranstaltung Übung zu Compiler Construction Project***

<b>Inhalte</b>	Students will write their own compiler using the functional language Haskell, which will consist of the following major components: scanner, parser, semantics analyser, (intermediate) code generator and code optimiser. Each student will be assigned a tutor (Betreuer) who will review the development of each component and help the student to reflect on her/his work.
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen

	Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	4
<b>Literatur</b>	<ul style="list-style-type: none"><li>• Louden, K. C. Compiler Construction: Principles and Practice. Course Technology, 1997.</li><li>• Aho, A. V., Sethi, R., Ullman, J. D. and Lam, M. S. Compilers: Principles, Techniques, and Tools, 2nd ed. Addison-Wesley, 2006.</li><li>• Fischer, C. N., Cytron, R. K. and LeBlanc Jr., R. J. Crafting a Compiler. Pearson, 2010.</li><li>• G. Hutton. Programming in Haskell. Cambridge University Press, 2007.</li></ul>
<b>Prüfungen</b>	Compiler Construction Project (Hausarbeit und Kolloquium)
	<b><i>Prüfung Compiler Construction Project (Hausarbeit und Kolloquium)</i></b>
<b>Beschreibung</b>	<ol style="list-style-type: none"><li>1) Documentation and submission of the compiler software that has been produced during the practicals (Übungen).</li><li>2) Questions concerning, and critical discussion of, the documented compiler software that has been produced during the practicals (Übungen).</li></ol>
<b>Typ</b>	Hausarbeit, Kolloquium
<b>Dauer</b>	20 Minuten

## Modul SWT-FSA-B: Foundations of Software Analysis (ab SoSe 2014)

<b>Modulgruppen</b>	A2 Allgemeine Informatik Grundlagen
<b>Lernziele / Kompetenzen</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 that are key for enhancing software quality. 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>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in diskreter Mathematik und Logik.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Mathematik für Informatiker 1 (Aussagen- und Prädikatenlogik) (GdI-MfI-1)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung  Der Aufwand von 180 Std. gliedert sich in etwa in: <ul style="list-style-type: none"><li>• 30 Std. Teilnahme an den Vorlesungen</li><li>• 30 Std. Teilnahme an den Übungen</li><li>• 60 Std. Vor- und Nachbereitung der Vorlesungen und Übungen, inkl. Recherche der Literatur und Studium zusätzlicher Quellen</li><li>• 30 Std. Bearbeitung der schriftlichen Hausarbeit</li><li>• 30 Std. Vorbereitung auf das Kolloquium</li></ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	

### ***Lehrveranstaltung Vorlesung zu Foundations of Software Analysis***

<b>Inhalte</b>	<p>Students will be introduced to the foundations of software analysis and their applications to software verification and code analysis and optimization. A particular emphasis will be put on semantics and abstraction, and their underlying mathematical theories.</p> <p>The following topics will be covered in the lectures (Vorlesungen): (i) semantics of programs; (ii) abstraction and abstract interpretation; (iii) elementary fixed point theory; (iv) operational and denotational abstract</p>
----------------	--

semantics; (v) software verification based on the methods of Floyd and Hoare; (vi) code analysis and optimization based on data flow analysis; (vii) outlook on advanced, modern aspects of software analysis.

<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<ul style="list-style-type: none"> <li>• Loeckx, J. and Sieber, K. The Foundations of Program Verification. 2nd ed. Wiley, 1987.</li> <li>• Further literature to be announced.</li> </ul>
<b>Prüfungen</b>	Foundations of Software Analysis (Hausarbeit und Kolloquium)

### ***Lehrveranstaltung Übung zu Foundations of Software Analysis***

<b>Inhalte</b>	The practicals (Übungen) will deepen the concepts and techniques taught in the lectures (Vorlesungen), and apply them to the analysis of small examples of software. They will cover not only pen-and-paper exercises, but 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, within the timetabled practicals (Übungen).
<b>Dozenten</b>	Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	- siehe gleichnamige Vorlesung -
<b>Prüfungen</b>	Foundations of Software Analysis (Hausarbeit und Kolloquium)

### ***Prüfung Foundations of Software Analysis (Hausarbeit und Kolloquium)***

<b>Beschreibung</b>	1) Pen-and-paper exercises practicing, reviewing and deepening the knowledge transferred in the lectures and practicals (Vorlesungen und Übungen).
---------------------	--

2) Questions testing the knowledge transferred in the lectures and practicals (Vorlesungen und Übungen), on the basis of the submitted solutions to the pen-and-paper exercises.

**Typ** Hausarbeit, Kolloquium

**Dauer** 20 Minuten

## Modul SWT-IPC-B: Imperative Programming Using C

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3W-I Wahlpflichtbereich I Schwerpunkt PuSWE
<b>Lernziele / Kompetenzen</b>	Students will develop an in-depth understanding of the C programming language, and acquire practical programming skills by learning how to develop clearly written and well-structured programs in ANSI C.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse der Programmierung und in Algorithmen und Datenstrukturen. Darüber hinaus sind Grundkenntnisse in Rechnerarchitekturen und Betriebssystemen wünschenswert.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Algorithmen und Datenstrukturen (MI-AuD-B)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung Arbeitsaufwand 90 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 30 Std. Teilnahme an den Übungen</li> <li>• 15 Std. Vor- und Nachbereitung der Übungen (einschließlich Lösen von Übungsaufgaben im Selbststudium)</li> <li>• 30 Std. Bearbeitung der schriftlichen Hausarbeit</li> <li>• 15 Std. Vorbereitung auf das Kolloquium</li> </ul>
<b>ECTS-Punkte</b>	3
<b>Bemerkung</b>	

### ***Lehrveranstaltung Übung zu Imperative Programming Using C***

<b>Inhalte</b>	The module covers the basic syntax of the C programming language, including types, operations and control structures. Concepts such as pointers, memory management, I/O handling and POSIX threads will be discussed in detail. Furthermore, it will be explained how the compiler, pre-processor, debugger, "make" tool and external libraries are employed. The practicals interleave this knowledge transfer with numerous examples and small programming tasks.
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen

<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<ul style="list-style-type: none"><li>• Kernighan, B. W. and Ritchie, D. The C Programming Language, 2nd ed. Prentice Hall, 1988.</li></ul>
<b>Prüfungen</b>	Imperative Programming Using C (Hausarbeit und Kolloquium)
<b><i>Prüfung Imperative Programming Using C (Hausarbeit und Kolloquium)</i></b>	
<b>Beschreibung</b>	<ol style="list-style-type: none"><li>1) Production and documentation of software in the C programming language, which has been written during the practicals (Übung).</li><li>2) Questions concerning the C programming language, and critical disussion of the documented software that has been written during the practicals (Übung).</li></ol>
<b>Typ</b>	Hausarbeit, Kolloquium
<b>Dauer</b>	20 Minuten



# Modul SWT-PCC-M: Principles of Compiler Construction

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3W-II Wahlpflichtbereich II Schwerpunkt PuSWE
<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. The module focuses on the theoretical and practical principles of compiler construction, which will enable students to gain 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; therefore, this module is a prerequisite for the module Compiler Construction Project (SWT-CCP-M).
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in den theoretischen Grundlagen der Informatik (speziell in Sprach- und Automatentheorie) und in Algorithmen und Datenstrukturen.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung  Arbeitsaufwand 180 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 45 Std. Teilnahme an den Vorlesungen</li> <li>• 30 Std. Nachbereitung der Vorlesungen, inkl. Recherche und Studium zusätzlicher Quellen</li> <li>• 15 Std. Teilnahme an den Übungen</li> <li>• 30 Std. Vor- und Nachbereitung der Übungen, inkl. Recherche und Studium zusätzlicher Quellen</li> <li>• 30 Std. Bearbeitung der schriftlichen Hausarbeit</li> <li>• 30 Std. Vorbereitung auf das Kolloquium</li> </ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	

***Lehrveranstaltung Vorlesung zu Principles of Compiler Construction***

<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, to dataflow analysis. The lectures will focus on the following aspects of compiler construction: lexical analysis, parsing, abstract syntax, semantic analysis, code generation and code optimisation.
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	3
<b>Literatur</b>	<ul style="list-style-type: none"><li>• Louden, K. C. Compiler Construction: Principles and Practice. Course Technology, 1997.</li><li>• Aho, A. V., Sethi, R., Ullman, J. D. and Lam, M. S. Compilers: Principles, Techniques, and Tools, 2nd ed. Addison-Wesley, 2006.</li><li>• Fischer, C. N., Cytron, R. K. and LeBlanc Jr., R. J. Crafting a Compiler. Pearson, 2010.</li></ul>
<b>Prüfungen</b>	Principles of Compiler Construction (Hausarbeit und Kolloquium)

### ***Lehrveranstaltung Übung zu Principles of Compiler Construction***

<b>Inhalte</b>	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 simultaneously attending the module Compiler Construction Project (SWT-CCP-M).
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich

<b>SWS</b>	1
<b>Literatur</b>	Siehe Vorlesung zu Principles of Compiler Construction
<b>Prüfungen</b>	Principles of Compiler Construction (Hausarbeit und Kolloquium)
<b><i>Prüfung Principles of Compiler Construction (Hausarbeit und Kolloquium)</i></b>	
<b>Beschreibung</b>	<p>1) Pen-and-paper exercises practicing, reviewing and deepening the knowledge transferred in the lectures (Vorlesungen) and practicals (Übungen).</p> <p>2) Questions testing the knowledge transferred in the lectures (Vorlesungen) and practicals (Übungen), on the basis of the submitted solutions to the pen-and-paper exercises.</p>
<b>Typ</b>	Hausarbeit, Kolloquium
<b>Dauer</b>	20 Minuten

## Modul SWT-PMI-B: Projektmanagement in IT-Projekten

<b>Modulgruppen</b>	A6 Kontextstudium->Allgemeine Schlüsselqualifikationen
<b>Lernziele / Kompetenzen</b>	<p>Ziel des Moduls ist es, den Teilnehmern eine wissenschaftlich und methodisch fundierte Erfahrung aus der Praxis der IT-Projekte zu vermitteln, orientiert an dem internationalen Standard ICB (IPMA Competence Baseline) der IPMA (International Project Management Association).</p> <p>Die Teilnehmer werden dadurch auch auf die Prüfung für ein Projektmanagement-Basiszertifikat, die selbst nicht Gegenstand dieses Moduls ist, der GPM (Deutsche Gesellschaft für Projektmanagement) vorbereitet.</p> <p>Arbeitsaufwand</p> <p>90 Std., welche sich grob wie folgt gliedern:</p> <ul style="list-style-type: none"><li>• 30 Std. Teilnahme an den Vorlesungen/Übungen</li><li>• 30 Std. Vor- und Nachbereitung der Vorlesungen/Übungen, inkl. Recherche und Studium zusätzlicher Quellen</li><li>• 30 Std. Vorbereitung auf die Klausur</li></ul>
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Das Modul richtet sich an Studierende ab dem 3. Semester im Kontextstudium des Bachelor-Studiengangs.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur
<b>ECTS-Punkte</b>	3
<b>Bemerkung</b>	
<b><i>Lehrveranstaltung Vorlesung/Übung zu Projektmanagement in IT-Projekten</i></b>	
<b>Inhalte</b>	Der Inhalt orientiert sich an den Anforderungen für die Erlangung des Projektmanagement-Basiszertifikats der GPM, nach dem aktuellen NCB (National Competence Guide). Es werden drei Kompetenzbereiche im Projektmanagement behandelt:

- Technische Kompetenz wie Projektanforderungen und Projektziele, Projektorganisation, Projektablauf und Termine
- Verhaltenskompetenz wie Führung, Kreativität, Umgang mit Konflikten und Krisen
- Kontextkompetenz wie Projektorientierung, Stammorganisation, Personalmanagement, rechtliche Aspekte

<b>Dozenten</b>	Dr. Sandra Bartsch-Beuerlein
<b>Sprache</b>	Deutsch
<b>Lehrformen</b>	Vorlesung und Übung (V/Ü)
<b>Häufigkeit</b>	WS, jährlich (jährlich im Wintersemester)
<b>SWS</b>	2
<b>Literatur</b>	<ul style="list-style-type: none"><li>• GPM (Hrsg.): ProjektManager, 2005</li><li>• GPM (Hrsg.): Kompetenzbasiertes Projektmanagement (PM3) , 2009</li><li>• GPM: NCB, National Competence Baseline Version 3.0, 2009</li><li>• Henrich, Andreas: Management von Softwareprojekten; Oldenbourg, 2002</li></ul>
<b>Prüfungen</b>	Projektmanagement in IT-Projekten (Klausur)
<b><i>Prüfung Projektmanagement in IT-Projekten (Klausur)</i></b>	
<b>Beschreibung</b>	Die Klausur prüft Wissen und Verständnis des in der Vorlesung/Übung vermittelten Lehrinhalts.
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

# Modul SWT-PMS-B: Modelle, Methoden und Werkzeuge für das Projektmanagement in Softwareprojekten

<b>Modulgruppen</b>	A6 Kontextstudium->Allgemeine Schlüsselqualifikationen
<b>Lernziele / Kompetenzen</b>	<p>Das Modul adressiert die spezifischen Probleme von Software-Entwicklungsprojekten, so werden zum Beispiel Vorgehensmodelle der Software-Entwicklung, Aufwandsschätzung für Softwareprojekte, iterativ-inkrementelle Planung von Softwareprojekten und weitere Methoden des Projektmanagements vermittelt. Ein besonderer Schwerpunkt liegt auf dem Einsatz von heute verfügbaren Werkzeugen des Projektmanagements in der Praxis.</p> <p>Arbeitsaufwand</p> <p>90 Std., welche sich grob wie folgt gliedern:</p> <ul style="list-style-type: none"><li>• 30 Std. Teilnahme an den Vorlesungen/Übungen</li><li>• 30 Std. Vor- und Nachbereitung der Vorlesungen/Übungen, inkl. Recherche und Studium zusätzlicher Quellen</li><li>• 30 Std. Vorbereitung auf die Klausur</li></ul>
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Das Modul richtet sich an Studierende ab dem 3. Semester im Kontextstudium des Bachelor-Studiengangs. Das Modul kann unabhängig vom Modul SWT-PMI-B "Projektmanagement in IT-Projekten" besucht werden; die Module SWT-PMS-B und SWT-PMI-B ergänzen sich dabei, können aber auch jeweils einzeln belegt werden.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur
<b>ECTS-Punkte</b>	3
<b>Bemerkung</b>	
	<b><i>Lehrveranstaltung Vorlesung/Übung zu Modelle, Methoden und Werkzeuge für das Projektmanagement in Softwareprojekten</i></b>
<b>Inhalte</b>	<ul style="list-style-type: none"><li>• Einführung und Überblick</li><li>• Projektorganisation</li><li>• Prozess- und Vorgehensmodelle in der Software-Entwicklung</li></ul>

- Projektstart
- Projektplanung
- Projektkontrolle und -steuerung
- Projektabnahme und -abschluss
- Qualitätssicherung
- Personalmanagement
- Risikomanagement
- Reifegradmodelle

**Dozenten** Dr. rer. nat. Karlheinz Morgenroth

**Sprache** Deutsch

**Lehrformen** Vorlesung und Übung (V/Ü)

**Häufigkeit** SS, jährlich

**SWS** 2

**Literatur**

- Hindel, Hörmann, Müller, Schmied: Basiswissen Software-Projektmanagement. dpunkt, 2006
- Burghardt: Projektmanagement. Wiley-VCH, 2000
- Frühauf, Ludewig, Sandmayr: Software-Projektmanagement und -Qualitätssicherung. vdf Hochschulverlag AG an der ETH Zürich, 1999
- Gartner, Wuttke: Projektmanagement - A Guide to the Project Management Body of Knowledge (deutsche Ausgabe des PMBOK), Westernacher, 2000
- Deutsche Gesellschaft für Projektmanagement (GPM): Projektmanagement Fachmann, Band 1 und Band 2, 6. Auflage, RKW Verlag, 2001
- Kerzner: Project Management. Wiley, 2001
- Sommerville: Software Engineering. Addison-Wesley, 1996
- Kruchten: The Rational Unified Process. Addison-Wesley, 1999
- Beck: eXtreme Programming Explained - Embrace Changes, Addison-Wesley, 2001
- Schwaber: Agile Project Management with Scrum. Microsoft Press, 2004
- V-Modell XT. [www.v-modell-xt.de](http://www.v-modell-xt.de)
- Royce: Software Project Management. Addison-Wesley, 1998
- Balzert: Lehrbuch der Software-Technik - Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum, 1998

**Prüfungen** Modelle, Methoden und Werkzeuge für das Projektmanagement in Softwareprojekten (Klausur)

***Prüfung Modelle, Methoden und Werkzeuge für das Projektmanagement in Softwareprojekten (Klausur)***

**Beschreibung** Die Klausur prüft Wissen und Verständnis des in der Vorlesung/Übung vermittelten Lehrinhalts.

**Typ** Klausur

**Dauer** 90 Minuten



# Modul SWT-SWE-B: Software Engineering

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3P Pflichtbereich Grundlagen PuSWT
<b>Lernziele / Kompetenzen</b>	Students will receive an introduction to the common problems, involving factors and paradigms in software development. They will also gather conceptual and practical knowledge, with an emphasis on requirements, analysis, design and testing of software.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in Informatik, Programmierkenntnisse in Java und Kenntnisse in Algorithmen und Datenstrukturen.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Klausur Arbeitsaufwand 180 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 30 Std. Teilnahme an den Vorlesungen</li> <li>• 45 Std. Nachbereitung der Vorlesungen, inkl. Recherche und Studium zusätzlicher Quellen</li> <li>• 30 Std. Teilnahme an den Übungen</li> <li>• 45 Std. Vor- und Nachbereitung der Übungen, inkl. Recherche und Studium zusätzlicher Quellen</li> <li>• 30 Std. Vorbereitung auf die Klausur</li> </ul>
<b>ECTS-Punkte</b>	6
<b>Bemerkung</b>	
<b><i>Lehrveranstaltung Vorlesung zu Software Engineering</i></b>	
<b>Inhalte</b>	The lectures (Vorlesungen) provide an introduction to software engineering, including commonly used processes, notations and techniques. All software engineering phases are discussed, with a focus on requirements, analysis, design and testing. In addition, specific aspects regarding flexible and agile development processes, software architectures, pattern-based development and software quality are presented.
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen
<b>Sprache</b>	Englisch/Deutsch

<b>Lehrformen</b>	Vorlesung (V)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	<ul style="list-style-type: none"><li>• Sommerville, I. Software Engineering, 9th ed. Addison-Wesley, 2010.</li><li>• Robertson, S. and Robertson, J. Mastering the Requirements Process, 2nd ed. Addison-Wesley, 2006.</li><li>• Stevens, P. and Pooley, R. Using UML - Software Engineering with Objects and Components. Addison-Wesley, 1999.</li><li>• Freeman, E., Freeman, 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. Addison-Wesley, 1994.</li><li>• Quatrani, T. Visual Modeling with IBM Rational Software Architect and UML. IBM Press, 2006.</li><li>• Zielcynski, P. Requirements Management using IBM Rational Requisite Pro. IBM Press, 2007.</li><li>• Tahchiev, P., Leme, F., Massol, V. and Gregory, G. JUnit in Action, 2nd ed. Manning Publications, 2010.</li></ul>
<b>Prüfungen</b>	Software Engineering (Klausur)

### ***Lehrveranstaltung Übung zu Software Engineering***

<b>Inhalte</b>	The practicals (Übungen) exercise and deepen the conceptual knowledge transferred via the lectures (Vorlesungen), and relay practical knowledge in software engineering.
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Siehe Vorlesung zu Software Engineering
<b>Prüfungen</b>	Software Engineering (Klausur)

### ***Prüfung Software Engineering (Klausur)***

<b>Beschreibung</b>	Written exam (Klausur) consisting of questions that relate to the contents of the lectures (Vorlesung) and practicals (Übung) of this module. The written exam is passed if at least 50% of the available points are reached.
<b>Typ</b>	Klausur
<b>Dauer</b>	90 Minuten

## Modul SWT-SWL-B: Software Engineering Lab

<b>Modulgruppen</b>	A3 Programmierung und Softwaretechnik->A3P Pflichtbereich Grundlagen PuSWT
<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>WWW</b>	-
<b>Arbeitsaufwand:</b>	180 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in Informatik und Softwaretechnik, Programmierkenntnisse in Java und Programmieren im Kleinen.
<b>Inhaltlich vorausgesetzte Module</b>	Modul Algorithmen und Datenstrukturen (MI-AuD-B) Modul Software Engineering (SWT-SWE-B)
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung Arbeitsaufwand 180 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"><li>• 25 Std. Teilnahme an Sitzungen des eigenen Teams zu Planung, Abstimmung und Feedback</li><li>• 20 Std. Teilnahme an den begleitenden Übungen (Tutorials) zu Softwarewerkzeugen</li><li>• 120 Std. Durchführung des Teamprojekts</li><li>• 15 Std. Erstellung der schriftlichen Hausarbeit und Vorbereitung auf das Kolloquium</li></ul>
<b>ECTS-Punkte</b>	6

### Bemerkung

### ***Lehrveranstaltung Übung zu Software Engineering Lab***

<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. Each team will also regularly meet with their tutor (a member of staff) in order to critically reflect on the
----------------	---

team's work, and participate in tutorials that introduce the software engineering tools to be used in this project.

<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Übung (Ü)
<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	4
<b>Literatur</b>	Siehe Modul Software Engineering (SWT-SWE-B)
<b>Prüfungen</b>	Software Engineering Lab (Hausarbeit und Kolloquium)

### ***Prüfung Software Engineering Lab (Hausarbeit und Kolloquium)***

<b>Beschreibung</b>	<p>1) Compilation of a written report (Schriftliche Hausarbeit) by each team, which must cover the following topics:</p> <ul style="list-style-type: none"> <li>• A description of the team's produced artefacts, including the electronic submission of the artefacts themselves;</li> <li>• A description, justification and critical reflection of the employed software engineering processes, methods and techniques in general and in each development phase;</li> <li>• A description of the team's organisation, the distribution of work and the contributions of each team member.</li> </ul> <p>The submission deadline and the details of the required content and format of this report will be announced at the beginning of the semester.</p> <p>2) Critical discussion of the team's produced software and software project report with respect to 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 technologies. The Kolloquium 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.</p> <p>Because this module involves a team effort, the examination can only be resit in a summer semester and requires active participation throughout the complete module in that semester.</p>
<b>Typ</b>	Hausarbeit, Kolloquium

**Dauer**

45 Minuten

## Modul SWT-TPL-B: Trends in Programming Languages (Bachelor)

<b>Modulgruppen</b>	A7 Seminare und Projekte->Seminare
<b>Lernziele / Kompetenzen</b>	Students will compile and acquire modern topics in programming languages by carrying out and documenting a guided literature survey and by preparing and delivering a coherent and comprehensible presentation to their peers.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in Programmiersprachen und im jeweils im Seminar behandelten Gebiet.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung Arbeitsaufwand 90 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 20 Std. Besprechungen und Vorträge (Referate) mit Diskussion</li> <li>• 25 Std. Literaturrecherche sowie Erarbeitung und Bewertung der Literatur</li> <li>• 30 Std. Anfertigen der schriftlichen Hausarbeit (Ausarbeitung)</li> <li>• 15 Std. Vorbereitung des Referats (Vortrags)</li> </ul>
<b>ECTS-Punkte</b>	3
<b>Bemerkung</b>	
<b><i>Lehrveranstaltung Seminar zu Trends in Programming Languages (Bachelor)</i></b>	
<b>Inhalte</b>	Various current topics in programming languages which complement and/or extend the technical and methodological aspects of the degree programme's foundational modules related to the field.
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch
<b>Lehrformen</b>	Seminar (S)

<b>Häufigkeit</b>	SS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Wird jeweils nach Seminarthemen vergeben
<b>Prüfungen</b>	Trends in Programming Languages (Bachelor) (Hausarbeit und Referat)

***Prüfung Trends in Programming Languages (Bachelor) (Hausarbeit und Referat)***

<b>Beschreibung</b>	1) Written report (Schriftliche Ausarbeitung) on the topic assigned to the student  2) Presentation (Vortrag) on the topic assigned to the student, including a discussion
<b>Typ</b>	Hausarbeit, Referat
<b>Dauer</b>	30 Minuten



## Modul SWT-TSE-B: Trends in Software Engineering (Bachelor)

<b>Modulgruppen</b>	A7 Seminare und Projekte->Seminare
<b>Lernziele / Kompetenzen</b>	Students will compile and acquire modern topics in software engineering by carrying out and documenting a guided literature survey and by preparing and delivering a coherent and comprehensible presentation to their peers.
<b>WWW</b>	-
<b>Arbeitsaufwand:</b>	90 Stunden
<b>Empfohlene Vorkenntnisse</b>	Grundlegende Kenntnisse in der Softwaretechnik und im jeweils im Seminar behandelten Gebiet.
<b>Inhaltlich vorausgesetzte Module</b>	-
<b>Bedingung für ECTS- Punkte</b>	Bestehen der Modulprüfung Arbeitsaufwand 90 Std., welche sich grob wie folgt gliedern: <ul style="list-style-type: none"> <li>• 20 Std. Besprechungen und Vorträge (Referate) mit Diskussion</li> <li>• 25 Std. Literaturrecherche sowie Erarbeitung und Bewertung der Literatur</li> <li>• 30 Std. Anfertigen der schriftlichen Hausarbeit (Ausarbeitung)</li> <li>• 15 Std. Vorbereitung des Referats (Vortrags)</li> </ul>
<b>ECTS-Punkte</b>	3
<b>Bemerkung</b>	
<b><i>Lehrveranstaltung Seminar zu Trends in Software Engineering (Bachelor)</i></b>	
<b>Inhalte</b>	Various current topics in software engineering which complement and/or extend the technical and methodological aspects of the degree programme's foundational modules, particularly the module Software Engineering (SWT-SWE-B).
<b>Dozenten</b>	Prof. Dr. Gerald Lüttgen Mitarbeiter Praktische Informatik, insbesondere Softwaretechnik und Programmiersprachen
<b>Sprache</b>	Englisch/Deutsch

<b>Lehrformen</b>	Seminar (S)
<b>Häufigkeit</b>	WS, jährlich
<b>SWS</b>	2
<b>Literatur</b>	Wird jeweils nach Seminarthemen vergeben
<b>Prüfungen</b>	Trends in Software Engineering (Bachelor) (Hausarbeit und Referat)
<b><i>Prüfung Trends in Software Engineering (Bachelor) (Hausarbeit und Referat)</i></b>	
<b>Beschreibung</b>	1) Written report (Schriftliche Ausarbeitung) on the topic assigned to the student 2) Presentation (Vortrag) on the topic assigned to the student, including a discussion
<b>Typ</b>	Hausarbeit, Referat
<b>Dauer</b>	30 Minuten