

**Dritte Ordnung
zur Änderung der Prüfungsordnung
für den Bachelorstudiengang Informatik
an der Fachhochschule Bielefeld
vom 17.06.2013**

Aufgrund des § 2 Abs. 4 und des § 64 Abs. 1 des Gesetzes über die Hochschulen des Landes Nordrhein-Westfalen (Hochschulgesetz - HG) vom 31. Oktober 2006 (GV. NRW. S. 474), zuletzt geändert durch Artikel 1 des Gesetzes vom 18. Dezember 2012 (GV. NRW. S. 672), hat der Fachbereich Technik der Fachhochschule Bielefeld die folgende Ordnung als Änderungssatzung erlassen:

Artikel I

Die Prüfungsordnung für den Bachelorstudiengang Informatik an der Fachhochschule Bielefeld vom 09.06.2011 (Verkündungsblatt der FH Bielefeld – Amtliche Bekanntmachungen – 2011, Nr. 18, Seite 747-859) in der Fassung der letzten Änderung vom 11.01.2013 (Verkündungsblatt der FH Bielefeld – Amtliche Bekanntmachungen – 2013, Nr. 3, Seite 6-7) wird wie folgt geändert:

§ 4 Anrechnung von Studienzeiten, Studien- und Prüfungsleistungen

Absatz 1:

Es wird geändert/ergänzt:

„Studienzeiten, Studien- und Prüfungsleistungen im gleichen Studiengang an anderen Hochschulen im Geltungsbereich des Grundgesetzes werden von Amts wegen angerechnet. Studien- und Prüfungsleistungen in anderen Studiengängen werden anerkannt, soweit die Gleichwertigkeit festgestellt ist. Gleichwertigkeit ist festzustellen, wenn

- Studienzeiten, Studien- und Prüfungsleistungen in Inhalt, Umfang und in den Anforderungen denjenigen des Bachelorstudiengangs an der Fachhochschule Bielefeld im Wesentlichen entsprechen (§ 63 Abs. 2 Satz 2 HG NRW) oder
- keine wesentlichen Unterschiede zwischen den zu vergleichenden Zeiten (Art. V Ziff. 1 Lissabon-Konvention) bzw. Leistungen (Art. VI Ziff. 1 Lissabon-Konvention) bestehen.

Dabei ist kein schematischer Vergleich, sondern eine Gesamtbetrachtung und Gesamtbewertung vorzunehmen.“

§ 15 Performanzprüfung

Absatz 4:

Es wird ergänzt:

„Die Gesamtnote ergibt sich als arithmetisches Mittel aus den Bewertungen der beiden bestandenen Einzelleistungen. Wenn eine Teilleistung endgültig nicht bestanden ist, gilt die gesamte Leistung als endgültig nicht bestanden.“

Studienplan

Wird gemäß Gegenüberstellung (siehe Anlage) geändert.

Das **Modulhandbuch** ist an den aktuellen Studienverlaufsplan angepasst und redaktionell überarbeitet worden.

Artikel II

Diese Ordnung wird im Verkündungsblatt der Fachhochschule Bielefeld – Amtliche Bekanntmachungen – bekannt gegeben. Sie tritt einen Tag nach ihrer Veröffentlichung in Kraft.

Ausgefertigt aufgrund eines Beschlusses des Fachbereichsrates des Fachbereichs Technik vom 22.05.2013.

Bielefeld, 17.06.2013

Die Präsidentin
der Fachhochschule Bielefeld

gez. Rennen-Allhoff

Prof. Dr. B. Rennen-Allhoff

Fachhochschule Bielefeld - Studiengang Informatik

Studienverlaufsplan (15.05.2013)

1. Semester	cps	SWS	V	SU	Ü	P
Einführung in das Berufsfeld	5	4	2		2	-
Mathematik für Informatiker I	8	6	3	-	3	-
Objektorientierte Programmierung	7	4	2	-	-	2
Skriptsprachen und XML	5	4	2	-	-	2
Technische Informatik	5	4	2	-	2	-
Summen	30	22	11	-	7	4
2. Semester	cps	SWS	V	SU	Ü	P
Mathematik für Informatiker II	8	6	3	-	3	-
Programmiermethoden	7	5	2	-	-	3
Algorithmen und Datenstrukturen	5	4	2	-	-	2
Theoretische Informatik	5	4	2	-	2	-
Useability und Datenvisualisierung	5	4		4	-	-
Summen	30	23	9	4	5	5
3. Semester	cps	SWS	V	SU	Ü	P
Software Engineering	7	4	2	-	-	2
Verteilte Systeme und Kommunikationsnetze	5	4	2	-	-	2
Datenbanken	5	4	2	-	-	2
Systemprogrammierung	8	6	3	-	-	3
Technical English	5	4	-	4	-	-
Summen	30	22	9	4	-	9
4. Semester	cps	SWS	V	SU	Ü	P
Softwareprojekt	5	4	-	-	-	4
Betriebssysteme	5	4	2	-	-	2
Embedded Systems	5	4	2	-	-	2
Computergraphik	5	4	2	-	-	2
Webbasierte Anwendungen	5	4	2	-	-	2
SW-Projektmanagement	5	4	-	4	-	-
Summen	30	24	8	4	-	12
5. Semester	cps	SWS	V	SU	Ü	P
Grundlagen der BWL	5	4	2	-	2	-
Wahlfach aus Liste 1	5	4	2	-	-	2
Wahlfach aus Liste 1	5	4	2	-	-	2
Wahlfach aus Liste 2	15	6	2	-	-	4
Summen	30	18	8	-	2	8
6. Semester	cps	SWS	V	SU	Ü	P
IT-Recht	5	4	-	4	-	-
Fachseminar	5	4	-	4		-
Wahlfach aus Liste 1	5	4	2	-	-	2
Wahlfach aus Liste 2	15	6	2	-	-	4
Summen	30	18	4	8	-	6
7. Semester	cps	SWS	V	SU	Ü	P
Praxisphase	18	-	-	-	-	-
Bachelorarbeit	12	-	-	-	-	-
Summen	30	-	-	-	-	-
Gesamtsummen	210	122	49	20	14	38

Fachhochschule Bielefeld - Studiengang Informatik
Studienverlaufsplan (15.05.2013)

Liste 1

„Methoden der Informatik“

(2 V, 2 P)

- Komponentenbasierte SW-Entwicklung
- Spezielle Methoden der Programmierung
- Mustererkennung & Bildverarbeitung
- Ausgewählte Konzepte von Datenbanksystemen
- Künstliche Intelligenz
- Verlässliche Systeme/Security
- Methoden der Computergraphik

Liste 2

„Anwendungen der Informatik“

(2 V, 4 P), Vertiefungsprojekt in Teamarbeit, von einem oder mehreren Professoren auch fachübergreifend betreut.)

- Anwendungen der Computergraphik
- Softwarequalität
- Webengineering
- Mobile Applikationen
- Softwareengineering für Embedded Systems
- Datenbankanwendungen
- Anwendungen der künstlichen Intelligenz

In jedem Semester stehen pro Liste mindestens 3 Wahlmodule zur Auswahl.

Im Vorfeld findet eine Bedarfsanalyse durch Befragung der Studierenden statt.

Um die Lehrkapazitäten effektiv einsetzen zu können, wird ein Wahlmodul nur angeboten, wenn sich mind. 7 Studierende dafür anmelden.

Die Fächervielfalt und Wahlmöglichkeiten wird gewährleistet, indem sich Listenfächer höchstens im Jahresrhythmus wiederholen.

Änderungen bleiben vorbehalten. Es gilt die jeweils aktuelle Fassung des Studienverlaufsplans.

Anlage 3: Modulhandbuch für den Bachelor-Studiengang Informatik

Inhaltsverzeichnis

Einführung in das Berufsfeld	186
Mathematik für Informatiker I	187
Objektorientierte Programmierung	188
Skriptsprachen und XML.....	189
Technische Informatik	190
Mathematik für Informatiker II.....	191
Programmiermethoden.....	192
Algorithmen und Datenstrukturen	193
Theoretische Informatik	194
Useability und Datenvisualisierung	195
Software Engineering.....	196
Verteilte Systeme und Kommunikationsnetze.....	197
Datenbanken.....	199
Systemprogrammierung.....	201
Technical English.....	203
Softwareprojekt.....	204
Betriebssysteme.....	205
Embedded Systems.....	205
Computergrafik	207
Webbasierte Anwendungen	209
SW-Projektmanagement	210
Grundlagen der BWL	211
Wahlfach aus Liste 1 "Methoden der Informatik" Komponentenbasierte SW-Entwicklung	212
Wahlfach aus Liste 1 „Methoden der Informatik“ Spezielle Methoden der Programmierung.....	214
Wahlfach aus Liste 1 "Methoden der Informatik" Mustererkennung & Bildverarbeitung.....	215
Wahlfach aus Liste 1 „Methoden der Informatik“ Ausgewählte Konzepte von DB-Systemen	216
Wahlfach aus Liste 1 „Methoden der Informatik“ Künstliche Intelligenz.....	217
Wahlfach aus Liste 1 "Methoden der Informatik" Verlässliche Systeme/Security	219
Wahlfach aus Liste 1 „Methoden der Informatik“ Methoden der Computergrafik	220
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Anwendungen der Computergrafik	221
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Software Qualität	222
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Webengineering.....	224
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Mobile Applikationen.....	225
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Softwareengineering für Embedded Systems	227
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Datenbankanwendungen.....	228
Wahlfach aus Liste 2 „Anwendungen der Informatik“ Anwendungen der künstlichen Intelligenz.....	229
IT-Recht	230
Fachseminar.....	231
Praxisphase	232
Bachelorarbeit.....	233

Einführung in das Berufsfeld					
Kenn-Nr. 1.1	Work-load 150 h	Credits 5 cps	Studien-semester 1. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Übung 2 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Es sollen die Wurzeln und Entwicklungsgeschichte der Informatik verstanden werden. Die Studierenden sollen die wechselseitigen Einwirkungen der Gesellschaft auf die Informatik und umgekehrt erkennen und selbst lernen, dazu Position zu beziehen. Die Studierenden sollen vielfältige erfolgreiche Informatiker und Ihren Berufsweg kennen lernen. Das Fachgebiet soll in seiner Vielfalt und Breite und den begrifflichen Teilgebieten verstanden werden. Die Studierenden sollen befähigt werden ihrem nachfolgenden Studium eine gezielte Fächerwahl zu treffen, unter Beachtung ihrer Stärken und Neigungen und im Hinblick auf ihr zukünftig angestrebtes Berufsfeld. Die Studierenden sollen in Ihrer Studienwahl bestärkt werden und für ein erfolgreiches Studium motiviert werden.				
3	Inhalte <ul style="list-style-type: none"> • Geschichte der Informatik; Errungenschaften der Informatik • Informatik und Gesellschaft • Teilgebiete der Informatik • Berufsbilder von Informatikern • Branchen und Bereiche für Absolventen der Informatik • Persönliche Entwicklungsperspektiven für Absolventen der Informatik 				
4	Lehrformen Seminar, Exkursionen				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens				
12	Sonstige Informationen -				

Mathematik für Informatiker I					
Kenn-Nr. 1.2	Work-load 240 h	Credits 8 cps	Studien-semester 1. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 3 SWS Übung 3 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 150 h	geplante Gruppengröße Vorlesung 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erwerben den korrekten Gebrauch der mathematischen Grundbegriffe und Grundlagen der Mengenlehre und Aussagenlogik. Sie beherrschen die wichtigsten Beweisverfahren und können Zählprinzipien der Kombinatorik anwenden. Es wird die Fähigkeit, mit Vektoren und Matrizen zu rechnen, sowie lineare Gleichungssysteme zu lösen, erlangt. Darüber hinaus erlernen sie den sicheren Umgang mit Funktionen und beherrschen die Differential- und Integralrechnung.				
3	Inhalte Grundlagen <ul style="list-style-type: none"> • Zahlen • Mengenlehre • Aussagenlogik • Vollständige Induktion • Kombinatorik Lineare Algebra <ul style="list-style-type: none"> • Vektoren und Vektorräume • Matrizen und lineare Abbildungen • Lineare Gleichungssysteme Analysis I <ul style="list-style-type: none"> • Folgen und Reihen • Reelle Funktionen einer Variablen • Differentialrechnung • Integralrechnung 				
4	Lehrformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 8/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller, Dipl. Inf. Birgit Christina George				
12	Sonstige Informationen Literatur Papula: Mathematik für Ingenieure Bronstein, Semendjajev: Taschenbuch der Mathematik				

Objektorientierte Programmierung					
Kenn-Nr.	Work-load	Credits	Studien-semester	Häufigkeit des Angebots	Dauer
1.3	210 h	7 cps	1. Sem.	Jedes WS	1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 150 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sollen befähigt werden, für einfache Problemstellungen objektorientierte Lösungen zu finden. Die Studierenden lernen den praktischen Umgang mit Programmierwerkzeugen. Selbst gefundene Lösungskonzepte können von den Studierenden modelliert, implementiert, dokumentiert und getestet werden. Die Studierenden lernen, Standardkomponenten und -bibliotheken in ihren eigenen Implementierungen zielgerichtet einzusetzen und zu nutzen. Es wird die Programmiersprache Java erlernt.				
3	Inhalte <ul style="list-style-type: none"> • Einführung in objektorientiertes Design • Einführung des Klassen- und Objektbegriffs • Nutzung einer Entwicklungsumgebung und eines Debuggers • Datentypen und Kontrollstrukturen • Einführung in die Vererbung, Polymorphie • Einführung in die Ausnahmebehandlung • Einführung von Collection-Klassen • Einführung in Swing • Ansätze zum Testen • Ansätze zum Dokumentieren (beispielsweise JavaDoc) 				
4	Lehrformen Vorlesung, Praktikum: die lt. Studienverlaufsplan 2 SWS Praktika werden durch 1 SWS tutorbetreutes Praktikum vor- bzw. nachbereitet				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistung der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 7/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips, N.N.				
12	Sonstige Informationen <ul style="list-style-type: none"> • Krüger, G.: "Handbuch der Java-Programmierung", Addison-Wesley, 2011 • Ullenboom, C.: "Java ist auch eine Insel", Galileo Computing, 2011 				

Skriptsprachen und XML					
Kenn-Nr. 1.4	Work-load 150 h	Credits 5 cps	Studien-semester 1. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS / 60 h	Selbst-studium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können <ul style="list-style-type: none"> • XML-Dokumente erstellen und verarbeiten • Sprachkonzepte von Skriptsprachen in eigenen Applikationen anwenden 				
3	Inhalte <ul style="list-style-type: none"> • Einführung: Bedeutung der XML-Datenformate, Bedeutung von Skriptsprachen • XML: Geschichte, Eigenschaften, Anwendungsbeispiele • Aufbau von XML-Datenformaten • Skriptsprachen: Anwendungsgebiete, Sprachkonzepte, z. B. Python • Erstellung von XML-basierten Programmoberflächen mithilfe von Skriptsprachen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen -				
6	Prüfungsvoraussetzungen -				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Dipl.-Inf. B.C. George				
12	Sonstige Informationen				

Technische Informatik					
Kenn-Nr. 1.5	Work-load 150 h	Credits 5 cps	Studien-semester 1. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Übung 2 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die grundlegenden Begriffe und Konzepte der technischen Informatik kennen lernen. <ul style="list-style-type: none"> • Grundbegriffe der technischen Informatik • Prinzipien der Computerarchitektur • grundlegende Kenntnisse über den Aufbau und die Funktionsweise von Digitalrechnern • Einführung einer hardwarenahen Programmiersprache (z.B. C, Assembler) 				
3	Inhalte <ul style="list-style-type: none"> • Grundlagen Rechnerarchitekturen <ul style="list-style-type: none"> ○ Einführung und Überblick ○ Aufbau von Digitalrechnern ○ Speicher und Peripherie ○ Bauformen und Anschlüsse • Grundlagen Digitaltechnik <ul style="list-style-type: none"> ○ Boolesche Algebra, Normalform ○ Schaltnetze ○ Schaltwerke • Hardwarenahe Programmierung <ul style="list-style-type: none"> ○ Entwicklungsumgebung, Cross-Compiler ○ Grundlagen der hardwarenahen Programmierung 				
4	Lehrformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Dipl.-Ing. Angela Kreienkamp (MPrComp)				
12	Sonstige Informationen Literatur: <ul style="list-style-type: none"> • z.B. Grundlagen der Technischen Informatik, Dirk W. Hoffmann • z.B. Technische Informatik: Eine einführende Darstellung, B. Becker, P. Molitor 				

Mathematik für Informatiker II					
Kenn-Nr. 2.1	Work-load 240 h	Credits 8 cps	Studien-semester 2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 3 SWS Übung 3 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 150 h	geplante Gruppengröße Vorlesung 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erlernen den sicheren Umgang mit Funktionen mehrerer Variablen und beherrschen die zugehörige Differential- und Integralrechnung. Sie erwerben ein Grundverständnis von linearen Differentialgleichungen. Relevante Zusammenhänge aus dem Bereich der Numerik sind bekannt, und elementare numerische Verfahren können durch die Studierenden angewandt werden. Im Rahmen der Veranstaltung werden die Studierenden befähigt, elementare stochastische Probleme mit geeigneten Methoden zu lösen. Darüber hinaus entwickeln die Studierenden ein Verständnis für wahrscheinlichkeitstheoretischer Grundbegriffe und kennen wichtige Verteilungen und ihre Bedeutung, sowie grundlegende statistische Methoden.				
3	Inhalte Analysis II <ul style="list-style-type: none"> • Lokale und globale Approximation • Differentialgleichungen • Reellwertige Funktionen mit mehrerer Variablen • Differential- und Integralrechnung für Funktionen mehrerer Variablen Numerik <ul style="list-style-type: none"> • Fehler und Fehlerfortpflanzung • Elementare numerische Verfahren • Optimierung Diskrete Wahrscheinlichkeitsrechnung und Statistik <ul style="list-style-type: none"> • Klassische Wahrscheinlichkeit • Bedingte Wahrscheinlichkeit • Zufallsgrößen • Verteilungen • Statistik 				
4	Lehrformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 8/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller, Dipl. Inf. Birgit Christina George				
12	Sonstige Informationen Literatur Papula: Mathematik für Ingenieure Bronstein, Semendjajev: Taschenbuch der Mathematik				

Programmiermethoden					
Kenn-Nr. 2.2	Work-load 210 h	Credits 7 cps	Studien-semester 2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 3 SWS, davon 1 SWS Tutor betreut	Kontaktzeit 5 SWS/75 h	Selbststudium 135 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen und beherrschen ein umfassendes Instrumentarium an Techniken und Lösungsmustern zur Softwareentwicklung in der Programmiersprache Java. Die Fähigkeit zum Entwickeln von eigenen Bibliotheken und komplexen Anwendungen in einzelnen Komponenten wurde erworben. Standardarchitekturmuster werden beherrscht.				
3	Inhalte <ul style="list-style-type: none"> • Vertiefung objektorientierter Entwurf/Design • Klassendiagramme mit UML • Einführung in Basis-Entwurfsmuster • fortgeschrittene Programmiermethoden (generische Programmierung, Reflection, RegExp, Logging, Konfiguration, Streams, Serialisierung, XML) • Annotationen und Enumerationen • Fehler- und Ausnahmebehandlung • Swing: Entwicklung komplexer GUIs, Java2D • Einführung in parallele Programmierung mit Java • systematischer Test, beispielsweise mit JUnit • Nutzung von Softwarebibliotheken (APIs) • Einführung in Versionsverwaltung • Einführung in Coding Conventions • Einführung Refactoring 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistung der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 7/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips				
12	Sonstige Informationen <ul style="list-style-type: none"> • Deitel, Deitel: "Java – How to Program", Pearson Education Limited, 2011 • Joshua Bloch: "Effective Java: A Programming Language Guide", Addison-Wesley, 2008 • Krüger, G.: "Handbuch der Java-Programmierung", Addison-Wesley, 2011 				

Algorithmen und Datenstrukturen					
Kenn-Nr. 2.3	Work-load 150 h	Credits 5 cps	Studien-semester 2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS / 60h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sollen einige wichtige klassische Algorithmen zum Sortieren und Suchen in verschiedenen Datenstrukturen kennenlernen. Sie sollen Erfahrungen sammeln, wie man Algorithmen schreiben, lesen und wie man ihren dynamischen Ablauf darstellen kann. Sie sollen üben, genau und verständlich über Algorithmen zu sprechen und zu schreiben. Sie sollen wichtige Fachbegriffe in Ihr aktives Vokabular aufnehmen. Sie sollen theoretisch und praktisch etwas über das Verhältnis von abstrakten Algorithmen und konkreter Programmierung erfahren. Sie sollen Ihre Fertigkeiten in qualitativ guter Programmierung in Java erweitern.				
3	Inhalte <ul style="list-style-type: none"> • Begrifflichkeiten (Algorithmus, Komplexität) • Gleichwertige Lösungen <ul style="list-style-type: none"> ○ Maximale Teilsumme ○ Komplexitätsformel ○ Datenstrukturen (Reihungen, verkettete Listen) • Rekursion und Wiederholung (Rekursive Algorithmen, Abarbeitung von Datenstrukturen) • Suchen in <ul style="list-style-type: none"> ○ Texten ○ Sammlungen ○ Reihungen ○ verketteten Listen ○ Hashtabellen • Sortierverfahren (quadratische, unterquadratische, rekursive, logarithmische) • Baumstrukturen (Suchen, Darstellen, Sortieren, Durchwandern, Operationen) 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Dipl.-Inf. B.C. George				
12	Sonstige Informationen Literaturangaben <ul style="list-style-type: none"> • Andreas Solymosi und Ulrich Grude: „Grundkurs Algorithmen und Datenstrukturen in Java“, Vieweg+Teubner, Wiesbaden 2008 • Ralf Hartmut Güting, Stefan Dieker: „Datenstrukturen und Algorithmen“, Vieweg +Teubner, Wiesbaden 2004 				

Theoretische Informatik					
Kenn-Nr. 2.4	Work-load 150 h	Credits 5 cps	Studien-semester 2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Übung 2 SWS	Kontaktzeit 4 SWS / 60h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können <ul style="list-style-type: none"> - zu gegebenen Automaten, Kellerautomaten, unterschiedlichen Arten von Grammatiken und Turingmaschinen die entsprechende Sprache angeben und umgekehrt - einen regulären Ausdruck in einen Automaten umwandeln und umgekehrt - die Chomsky-Hierarchie aufstellen und den Klassen Sprachen zuordnen - die Probleme der Berechenbarkeit und der Entscheidbarkeit und das Halteproblem diskutieren - das P-NP-Problem an Hand von Beispielen diskutieren - die Phasen des Compilers nennen und dabei die Anwendung von Automaten und Grammatiken erklären 				
3	Inhalte <ul style="list-style-type: none"> - Automaten (mit Ausgabe, deterministisch, nicht deterministisch) - reguläre Ausdrücke - Grammatiken, kontextfreie Sprachen - Kellerautomaten - kontextsensitive und Typ 0 - Sprachen, Turingmaschine - Chomsky-Hierarchie - Berechenbarkeit, Entscheidbarkeit, Church'sche These, Halteproblem - Komplexitätstheorie, P-NP-Problem - Compilerbau 				
4	Lehrformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Dipl.-Inf. Ludger Franzen				
12	Sonstige Informationen Hopcroft, Ullman, Motwani, Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson, 2002 Schöning, Theoretische Informatik - kurz gefasst, Spektrum, 2008 Michael Sipser, Introduction to The Theory of Computation, Thomson Press, 2005				

Useability und Datenvisualisierung					
Kenn-Nr. 2.5	Work-load 150 h	Credits 5 cps	Studien-semester 2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS	Kontaktzeit 4 SWS / 60h	Selbststudium 90 h	geplante Gruppengröße Seminaristischer Unterricht 35	
2	Lernergebnisse (learning outcomes) / Kompetenzen Studierende lernen die Grundbegriffe von Useability Engineering kennen. Sie kennen die gängigen Useability Engineering Methoden und Praktiken. Darüber hinaus können sie ausgewählte Engineering-Methoden anwenden, indem sie einen besonderen Wert auf die Userverhalten und -anforderungen legen. Die Studierenden erlernen Techniken einer nutzerzentrierten Datenvisualisierung.				
3	Inhalte <ul style="list-style-type: none"> • Einführung: Definitionen, Grundbegriffe • Grundlagen von Benutzerschnittstellen • Gestaltungsrichtlinien • Vorgehensmodelle und Methoden des Useability Engineering • Technische Rahmenbedingungen • Gestaltungsprinzipien, Werkzeuge • Wahrnehmungspsychologische Aspekte der Datenvisualisierung • Techniken der Datenvisualisierung • Graphische Userinterfaces 				
4	Lehrformen Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Seminaristischen Unterricht mit Testat Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller				
12	Sonstige Informationen				

Software Engineering					
Kenn-Nr. 3.1	Work-load 210 h	Credits 7 cps	Studien-semester 3. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 150 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erlernen ein systematisches und ingenieurmäßiges Vorgehen zur erfolgreichen Planung und Durchführung eines Softwareentwicklungsprojektes von der Anforderungsanalyse über Grobdesign, Feindesign, Implementierung bis hin zur Qualitätssicherung. Sie erlernen Standards und Tooleinsatz, und den gezielten Einsatz von UML. Die Programmierkenntnisse in Java werden erweitert u.a. um Asserts, JUnit - tests, Pattern, Frameworks und komponentenbasiertes Design.				
3	Inhalte <ul style="list-style-type: none"> • Einführung in Softwareengineering • Prozessmodellierung (Unternehmensprozesse, Aktivitätsdiagramme, Risikomanagement) • Vorgehensmodelle (Wasserfallmodell, Prototypische Entwicklung, Iterative Entwicklung, Iterativ-inkrementelle Entwicklung, Allgemeines V-Modell, Agile Entwicklung wie Scrum und Extreme Programming) • Anforderungsanalyse (Stakeholder, Ziele, Uses Cases, Ableitung funktionaler Anforderungen, nicht funktionale Anforderungen, Lasten- und Pflichtenheft) • Grobdesign (Systemarchitektur, Ableitung der grundlegenden Klassen, Methoden, Sequenzdiagramm, Überlegungen zur Oberflächenentwicklung) • Programmgenerierung(CASE-Tools, Übersetzung von Klassen und Assoziationen, Arten der Objektzugehörigkeit, Software-Architektur) • Feindesign (Details im Kleinen, Model View Controller, GoF-Pattern) • Implementierungen (Verteilte Systeme, XML, Programmbibliotheken, Komponenten, Frameworks, Persistente Datenhaltung, Model Driven Architecture) • Oberflächengestaltung (Useability, Ergonomie, Prüfung) • SW-Qualitätssicherung (Zusicherungen, Unit-Tests, Testverfahren, Testarten, Metriken) 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 7/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens, N.N.				
12	Sonstige Informationen Literaturangaben: Stephan Kleuker: „Grundkurs Software Engineering mit UML: Der pragmatische Weg zu erfolgreichen Softwareprojekten“, Vieweg&Teubner, Wiesbaden 2010 Bernd Oesterreich: "Analyse und Design mit UML" Oldenbourg, München 2004				

Verteilte Systeme und Kommunikationsnetze					
Kenn-Nr.	Work-load	Credits	Studien-semester	Häufigkeit des Angebots	Dauer
3.2	150 h	5 cps	3. Sem.	Jedes WS	1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Kommunikation über Netze und das Internet bilden einen grundlegenden Bestandteil der modernen Berufswelt. Darüber hinaus sind Ethernet-Technologien und TCP/IP-basierte Kommunikation ein elementarer Bestandteil der meisten verteilten informationstechnischen Systeme.</p> <p>Die Studierenden erlernen grundlegende Kenntnisse digitaler Kommunikationsnetze mit dem Schwerpunkt Rechnerkommunikation.</p> <p>Sie lernen die Architektur und wichtige Methoden und Werkzeuge für verteilte Systeme kennen. Sie entwickeln die Fähigkeit, eigenständige verteilte Systeme und Netzwerk-Anwendungen zu entwickeln.</p> <p>Sie lernen das Bewerten der Stärken und Schwächen unterschiedlicher Ansätze für verteilte Anwendungen.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Grundlagen der Kommunikationsnetze <ul style="list-style-type: none"> - Kommunikationsmodelle/-protokolle und Standards - Vermittlungsprinzipien • Technologien für lokale Netze (LAN) <ul style="list-style-type: none"> - Übertragungsmedien - Stochastische und deterministische Medienzugriffsverfahren - Ethernet-Technologien und Protokolle • Protokolle der TCP/IP-Protokollfamilie (IP, ICMP, UDP, TCP, Protokolle der Anwendungsschicht) • Übersicht über Technologien für Weitverkehrsnetze • Grundlagen IP-basierter Routing-Protokolle • Router und Routerkonfiguration • Routing-Protokolle • Funktionsorientierte Client/Server-Systeme mit Remote Procedure Calls (RPC) • Objektorientierte, verteilte Systeme (CORBA, Java RMI etc.) • Sicherheit von Verteilten Anwendungen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine.				
6	Prüfungsvoraussetzungen Keine.				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Martin Hoffmann				

12**Sonstige Informationen**

Bengel, Günther: Verteilte Systeme, Client-Server-Computing für Studenten und Praktiker, Vieweg-Verlag Braunschweig/Wiesbaden, 2. Auflage 2002.

Comer, Douglas E.: Computernetzwerke und Internets, Prentice Hall/Pearson Studium, 2002

U. Hammerschall: Verteilte Systeme und Anwendungen : Architekturkonzepte, Standards und Middleware-Technologien, München [u.a.] : Pearson Studium, 2005.

Alexander Schill, Thomas Springer: Verteilte Systeme - Grundlagen und Basistechnologien, 368 Seiten,

Springer, 2009, ISBN: 3540205683.

Datenbanken					
Kenn-Nr. 3.3	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Datenbanken sind eine Basistechnologie der Informatik. Ziel der Veranstaltung ist ein sicheres Verständnis der grundlegenden Konzepte und Sprachen von Datenbanksystemen zu vermitteln.</p> <p>Die Studierenden wenden ein einheitliches konsistentes Begriffsgebäude bezüglich der Datenbankthematik an. Sie können die theoretischen Grundlagen von Datenbanksystemen an Beispielen erläutern, insbesondere funktionale Abhängigkeiten, relationale Algebra und Normalisierung, sie wissen um die Aufgaben und den Sinn der grundlegenden Architektur von DBMS. Sie können die Dokumentation mehrerer wichtiger DBMS gezielt problemlösend durchsuchen und kennen die grundlegenden Funktionen der Clients mehrerer DBMS und nutzen Sie zur Kommunikation und Programmierung von Datenbanken. Sie modellieren komplexe Sachverhalte sicher in auch umfangreiche Datenmodelle und implementieren sie in verschiedenen DBMS. Sie fällen dabei begründete Entscheidungen für die Anwendungen von Constraints, Domänen und Datentypen. Sie wenden SQL sicher zur Lösung komplexer Informationsbedarfe an und erstellen umfangreiche nichttriviale Abfragen. Dabei verwenden sie sowohl den aktuellen SQL Standard (zur Zeit SQL:2011) als auch die Dialekte mehrerer wichtiger DBMS. Sie verstehen den Transaktionsbegriff, beschreiben Probleme/Phänomene der Mehrbenutzersynchronisation und Nebenläufigkeit in Read/Write Notation, und entscheiden, wie man sie durch Isolation von Transaktionen verhindert – sowohl durch Standard-Isolationslevel, als auch durch spezifische Implementierungen in mehreren DBMS. Sie greifen über Datenbankschnittstellen aus eigenen Programmen auf Datenbanken zu und verarbeiten Datensätze in Programmen und Datenbanken. Sie programmieren Persistent Stored Modules in einem der besprochenen DBMS.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Aufgaben und Architektur von Datenbanksystemen • Clients und Schnittstellen zu Datenbanksystemen • Grundlagen des relationalen Modells • E/R Modellierung, logische und physische Datenmodelle, SQL Datentypen, Implementierung in wichtigen DBMS • Constraints, Assertions, Integrität, Domänen, Datentypen • SQL:2011, insbesondere SQL-schema statements, SQL-data statements, SQL-data change statements, SQL-transaction statements und SQL-connection statements • Transaktionskonzepte, Nebenläufigkeit, Isolationslevel • Datenbankschnittstellen (JDBC, ODBC) • Grundlagen von Persistent Stored Modules, Programmierung von PSM, Trigger 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) Keine
10	Stellenwert der Note für die Endnote 5/210
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking
12	Sonstige Informationen <ul style="list-style-type: none">• Kleuker, S., Grundkurs Datenbankentwicklung, Vieweg Teubner, 2011• Kemper, A, Eickler, A, Datenbanksysteme - Eine Einführung, Oldenbourg, 2011• Piepmeyer, L., Grundkurs Datenbanksysteme, Hanser, 2011• Saake, S., Sattler, K.-U., Heuer, A., Datenbanken - Konzepte und Sprachen, mitp, 2010

Systemprogrammierung					
Kenn-Nr. 3.4	Work-load 240 h	Credits 8 cps	Studien-semester 3. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 3 SWS Praktikum 3 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 150 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Systemnahe Software bildet eine Abstraktionsschicht zwischen dem Betriebssystem und den Anwendungen. Unter Nutzung der einfach gehaltenen Betriebssystemschnittstellen soll diese Schicht dem Anwendungsentwickler effiziente und robuste Schnittstellen zur Verfügung stellen, die durch den höheren Abstraktionsgrad deutlich einfacher als die Betriebssystemschnittstellen anzuwenden sind. Systemprogrammierung ist damit ein Schlüssel zum Verständnis der internen Abläufe in einem IT-System.</p> <p>Nach erfolgreichem Abschluss dieser Lehrveranstaltung haben die Studierenden erworben:</p> <ul style="list-style-type: none"> • Kompetenzen zur Entwicklung systemnaher Software in einer Systemprogrammiersprache, beispielsweise C/C++ • Verständnis der Zusammenhänge, die die Ausführung von Anwendungen in vielschichtig organisierten Rechensystemen ermöglichen • Vertieftes Wissen über die Systemaufrufe eines konkreten Systems (UNIX/Linux oder Windows) und deren Anwendung in der Praxis • Programmentwicklung unter Nutzung wesentlicher Teile der UNIX-Programmierschnittstelle (API) beherrschen • Fähigkeiten auch die internen Schnittstellen eines Betriebssystems zu nutzen, um eigene Komponenten (z.B. Treiber) integrieren zu können • Die Kompetenz dieses Wissen dann selbständig auf größere Aufgabenstellungen anwenden zu können (Praktikumsaufgaben) 				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Grundlagen (Begriff der Systemprogrammierung, Systemaufrufe, Systemprogrammierungssprachen, Beispiele, Portabilität, Programmierstil) • Konkrete Systeme (z.B. UNIX/Linux, Windows) • Einführung in die Systemprogrammierungssprachen C und C++ • Modulare Programmierung, Präprozessormakros und Automatisierung von Abläufen • Umgang mit statischen und dynamischen Bibliotheken • Metaprogrammierung mit Templates • Einführung in Standardbibliotheken (z.B. STL, Boost) • Fehlerbehandlung (System, Programmiersprache) • Ein- und Ausgabe mit Hilfe von C++-Streams • Speicherverwaltung, dynamischer Speicher, Pointer • Prozess- und Thread-Manipulation (Erzeugen, Beenden, Zustände) • Interprozess-Kommunikation (Überblick, Signale, Sockets) • aktuelle Systemabstraktionen, beispielsweise D-Bus • Einführung in QT und QML • Sichere und defensive Programmierung • Wechselnde Inhalte der Praktika zu aktuellen Themen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistung der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

	-
10	Stellenwert der Note für die Endnote 8/210
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips
12	Sonstige Informationen <ul style="list-style-type: none">• Ulrich Breymann: "Der C++ Programmierer", Hanser, 2011.• Love, Oram: "Linux System Programming", O'Reilly Media, 2007.• Michael Kerrisk: "The Linux Programming Interface", no starch press, 2011.

Technical English					
Kenn-Nr. 3.5	Work-load 150 h	Credits 5 cps	Studien-semester 3. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Seminaristischer Unterricht 35	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Teilnehmer sollen in der Lage sein sich mit technischem Englisch auszudrücken. Sie sollen diese erworbenen Sprachkenntnisse in einem internationalen Umfeld sowohl schriftlich als auch gesprochen anwenden können. Die Studierenden erlernen hierzu die spezifische Terminologie, welche im Bereich Informatik angewendet wird, als auch allgemeines technisches Englisch und werden in der Lage sein Prozesse zu beschreiben und zu erklären, technische Probleme zu diskutieren und IT-Lösungen zu vermitteln.				
3	Inhalte <ul style="list-style-type: none"> • Computer Hardware: Ein- und Ausgabemedien, Drucker, Speichermedien • Wesentliche und kreative Software: Betriebssystem, Textverarbeitung, Tabellenkalkulation und Datenbanken, Desktop-Publishing, Multimedia • Programmierung : Programmiersprachen, Java, Webgestaltung • Zukunftstechnologien: Kommunikationssysteme, Netzwerke, Videospiele • Internet: E-Mail, das Web, Videokonferenzführung • Beschäftigungsmöglichkeiten im Bereich IKT 				
4	Lehrformen Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Seminaristischen Unterricht mit Testat Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Cathrine Stones				
12	Sonstige Informationen				

Softwareprojekt

Kenn-Nr. 4.1	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Praktikum 4 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen In enger Verzahnung mit dem Modul "Projektmanagement" Die Studierenden wenden die wesentlichen Grundlagen des Projektmanagements an. Sie planen ein größeres Softwareprojekt, führen es durch, leiten es und dokumentieren und präsentieren regelmäßig Projektfortschritt und -ergebnisse. Sie entscheiden sich begründet in dem gegebenen Projektzusammenhang für ein Vorgehensmodell. Sie wenden Vorgehensweise, Organisationsformen und Methoden eines anerkannten formalen Projektmanagementsystems auf ihr Projekt an. Sie realisieren ein Projekt in einer größeren Projektgruppe (ca. 8 Personen) mit Rollenverteilung.				
3	Inhalte <ul style="list-style-type: none"> • Einrichtung und Durchführung eines Projektes • Erstellung eines Pflichtenhefts auf Basis der Vorgaben eines fiktiven „Kunden“, Durchführung einer Aufwandsschätzung auf Basis dieses Pflichtenhefts • Aufstellen eines Projektplanes und Verfahren zur Projektverfolgung und des Risikomanagements • Implementierung im Team (Version-Management, Build-Verfahren, Abstimmungsprozesse, Schnittstellen) • Präsentation von Ergebnissen und Zwischenergebnissen • Einsatz aktueller Technologien zur Implementierung der Anwendung • Abschlusspräsentation 				
4	Lehrformen Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende N.N., Prof. Dr. Becking, Prof. Dr. Behrens, Prof. Dr. Gips, Prof. Dr. Hoffmann				
12	Sonstige Informationen				

Betriebssysteme					
Kenn-Nr. 4.2	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Betriebssysteme verwalten die Ressourcen eines Rechnersystems und bestimmen dadurch wesentlich dessen Leistungsfähigkeit und Nutzbarkeit. Gute Kenntnisse wesentlicher Betriebssystem-Konzepte sind daher für das Verständnis moderner IT Systeme unerlässlich. Studierende besitzen nach Abschluss der Lehrveranstaltung die folgenden Fähigkeiten: - Systembezogene Abstraktionen kennen, verstehen und wiedererkennen, wie sie von Betriebssystemen verwendet und bereitgestellt werden - Wichtige Verfahren und Algorithmen aus dem Bereich Betriebssysteme sicher anwenden - Methoden kennen und verstehen, die bei der Konstruktion von Betriebssystemen Verwendung finden - beides an Beispielsystemen detailliert erläutern können (primär UNIX/Linux z.T. Windows)				
3	Inhalte <ul style="list-style-type: none"> • Einführung: Aufgaben, Arten und Strukturen von Betriebssystemen • Kurzvorstellung aktuelle Rechnerarchitekturen (Pipeline, Cache, Branch Prediction) • Nebenläufigkeit, Prozessverwaltung, Scheduling • Synchronisation und Kommunikation • Speicherverwaltung: Virtueller Speicher, Segmentierung, Paging, Seitenersetzungsalgorithmen • Ein-/Ausgabe: zeichen- und blockorientierte Geräte • Dateisysteme: Beispiele, Aufbau, Fehlertoleranz • Treiber-Modelle und -Programmierung • Sicherheit: Authentifikation, Schutzmechanismen, Autorisierung, vertrauenswürdige Systeme • Wechselnde Inhalte der Praktika zu aktuellen Themen (z.B. Bash-Skripting, Entwicklung eigener Shell, Programmierung Linux Kerneltreiber) 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistung der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Martin Hoffmann				
12	Sonstige Informationen <ul style="list-style-type: none"> • Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, 2009 • Stallings: "Operating Systems: Internals and Design Principles", Prentice Hall, International, 2011 				

Kenn-Nr. 4.3	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erlernen die grundlegenden Kenntnisse zur Umsetzung von eingebetteten Systemen (embedded systems). Sie erhalten ein Verständnis der Besonderheiten des Entwurfs eingebetteter Systeme und der zur Implementierung eingebetteter Systeme benötigten Kompetenzen. Insbesondere werden den Studierenden als "Handwerkszeug" die entsprechenden Spezifikations- und Programmieretechniken, Modelle der Ablaufplanung und Software- und Systemarchitekturen für eingebettete Systeme mitgegeben. Den Studierenden werden anhand von exemplarischen Anwendungsszenarien die typischen Entwurfsschritte vermittelt, um Software für eingebettete Systeme zu entwickeln (Praktikum).				
3	Inhalte <ul style="list-style-type: none"> • Überblick und Anwendungsgebiete • Besonderheiten und Anforderungen • Zusammenspiel Software und Hardware • Entwurfsschritte • Spezifikation und Modellierung • Realisierung und Implementierung • Validierung und Evaluation Praktische Umsetzung eingebetteter Systeme an Anwendungsbeispielen.				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: P. Marwedel, Eingebettete Systeme, Springer 2008				

Computergrafik					
Kenn-Nr. 4.4	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erlernen die fundamentalen Methoden und Algorithmen in der Computergrafik. Grundlegende Begriffe werden eingeführt und sowohl Verfahren aus der 2D wie auch 3D Computergrafik erläutert. Ebenso werden die mathematischen Grundlagen für die 2D und 3D Computergrafik erworben und in Anwendungsbeispielen eingeübt. Die Studierenden kennen gängige Tools und Anwender-Werkzeuge aus der Computergrafik und sind mit den zugehörigen Technologien vertraut. Sie sind in der Lage mit einem aktuellen Tool eine virtuelle Szene zu modellieren und zu rendern.				
3	Inhalte <ul style="list-style-type: none"> • Historie, Überblick, Beispiele: Anwender-Werkzeuge, Tools, Anwendungsbeispiele. • Begriffe und Grundlagen: Grafische Eingabegeräte, Bildschirmtechnologien, 3D-Sichtsysteme, Rastergrafik. • Objekt- und Sichttransformationen: Koordinatensysteme, Transformationen, Projektionen, Clipping. • Repräsentation und Modellierung von Objekten: Polygonale Repräsentation, Raumteilungsverfahren, Szenenbeschreibung. • Rendering und Visibilität: Farbmodelle, Visibilitätsverfahren, Beleuchtung und Schattierung, lokale Beleuchtungsmodelle, interpolative Schattierungstechniken, globale Beleuchtungsmodelle, Rendering-Pipelines. • Tools zur Modellierung/Rendering: Modellierung und Rendern einer kleinen Szene mit Hilfe z.B. von Autodesk Maya, Blender. 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller				

12	Sonstige Informationen Literaturangaben: <ul style="list-style-type: none">• Bender M., Brill, M.: Computergrafik, 2. Auflage, Hanser Verlag, 2005. http://www.vislab.de• Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition.• Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics – Principles and Practice, Addison-Wesley
-----------	---

Webbasierte Anwendungen					
Kenn-Nr. 4.5	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Webbasierte Systeme sind eine der verbreitetsten Ausprägungen verteilter Informationssysteme und sind heute weder aus dem privaten noch aus dem beruflichen Bereich wegzudenken. Solche Systeme professionell planen, realisieren und beurteilen zu können, stellt daher eine wichtige Qualifikation dar. <ul style="list-style-type: none"> • Kennen typischer Anwendungsgebiete Web-basierter Anwendungen • Fähigkeit zum problemadäquaten Entwurf und zur Realisierung Web-basierter Anwendungen, einschl. Beurteilung und Auswahl geeigneter Basistechnologien 				
3	Inhalte <ul style="list-style-type: none"> • Einführung, Klassifikation von Web-Anwendungen, Architekturen • Grundlagen (HTTP, Sessionmanagement, Standardisierung, W3C) • Sicherheit von Webanwendungen • Webseiten gestalten (XHTML, Cascading Stylesheets, HTML5) • Webuseability (Suchmaschinenoptimierung, Metriken auswerten, Besucherverhalten verstehen, Websites optimieren) • Clientseitige Technologien: JavaScript, Ajax, DOM • Serverseitige Technologien: PHP, Servlets, JSP's • Webservices: SOAP, WSDL • Mehrschichtenarchitekturen, Frameworks zu deren Umsetzung: JSF, Struts • Applikationsserver (Aufgaben, Dienste, Beispiele): Apache Tomcat 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens				
12	Sonstige Informationen Literatur: Bernd Müller: „JavaSever Faces 2.0: Ein Arbeitsbuch für die Praxis“, Hanser 2010 Martin Marinschek: „JavaServer Faces 2.0: Grundlagen und erweiterte Konzepte“, dpunkt 2009 Sven Haiges: „HTML5-Apps für Iphone und Android“, Franzis 2011 Stefan Koch: „Javascript: Einführung, Programmierung und Referenz“, dpunkt 2011				

SW-Projektmanagement					
Kenn-Nr. 4.6	Work-load 150 h	Credits 5 cps	Studien-semester 4. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Seminaristischer Unterricht 35	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die wesentlichen Grundlagen des Projektmanagements. Sie sind in der Lage, Projekte zu planen, durchzuführen, in Teilen zu leiten sowie Projektfortschritt und -ergebnisse zu dokumentieren und zu präsentieren. Sie können sich begründet in einem Projektzusammenhang für ein Vorgehensmodell entscheiden. Sie kennen Vorgehensweise, Organisationsformen und Methoden eines anerkannten formalen Projektmanagementsystems.				
3	Inhalte <ul style="list-style-type: none"> • Projekte als Problemlösungsprozesse • Gründung, Organisation und Strukturierung von Projekten • Projektplanung • Projektsteuerung • Management von Software-Projekten • Werkzeuge im Projektmanagement • Kommunikation und Dokumentation als Querschnittsaufgabe • Qualitätssicherung • Projektmanagementsysteme 				
4	Lehrformen Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Seminaristischen Unterricht mit Testat Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking				
12	Sonstige Informationen				

Grundlagen der BWL					
Kenn-Nr. 5.1	Work-load 150 h	Credits 5 cps	Studien-semester 5. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Übung 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erwerben Wissen um betriebswirtschaftliche Zusammenhänge im beruflichen Umfeld eines Informatikers. Sie kennen wesentliche betriebswirtschaftliche Verfahren und Grundbegriffe, haben einen Überblick zu Rechtsformen von Unternehmen, zur Investition und Finanzierung und zur Produktionsplanung und -steuerung. Sie haben einen Überblick zu Marketingstrategien.				
3	Inhalte <ul style="list-style-type: none"> • Grundbegriffe der BWL • strategische Ausrichtung des Unternehmens (Portfolio-Analyse, Wettbewerb), • Investition und Finanzierung, • Betriebsorganisation • Controlling • Marketing • Personalwesen • Personalführung unter Berücksichtigung von Führungsstrukturen • Einsatz moderner Softwaretools 				
4	Lehrformen Vorlesung, Übung				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Hausarbeit, mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Michael Mohe, Lehrbeauftragte				
12	Sonstige Informationen				

Wahlfach aus Liste 1 "Methoden der Informatik"

Komponentenbasierte SW-Entwicklung

Kenn-Nr. 5.2	Workload 150 h	Credits 5 cps	Studien- semester 5. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Studierende sollen komponentenbasierende Software-Entwicklung nutzen lernen. Der Vergleich zur objektorientierten Software-Entwicklung soll insbes. in Bezug auf die losere Kopplung und die verteilte Nutzbarkeit erkannt werden. Marktübliche Frameworks und deren Möglichkeiten sollen kennen gelernt werden. Die Studierenden verfügen über Kenntnisse über Frameworks der komponentenbasierten Software-Entwicklung. Die Studierenden sind in der Lage die Möglichkeiten marktüblicher Frameworks zu bewerten und diese entsprechend der Anforderungen einzusetzen. Die Studierenden können komponentenbasierte Frameworks zur Implementation von Anwendungen einsetzen.				
3	Inhalte 1. Einführung und Geschichte COM/DCOM, DLLs 2. Vergleich Objektorientierte/komponentenbasierende Software-Entwicklung 3. Applikationsserver 4. Grundlegende Dienste von Komponentenstandards 4.1 Verteilung, Erzeugung, Aktivierung Deaktivierung von Komponenten 4.2 Naming/Verzeichnisdienste 4.3 Messaging 4.4 Transaktionen 4.5 Sicherheit/Redundanz 4.6 Persistenz, Datenbankschnittstellen 4.7 Weitere Dienste 5 Übersicht aktuelle Komponentenstandards 5.1 CCM (Corba 3.0) 5.2 COM+ 5.3 EJB (J2EE) 6. Nutzung aktueller Komponentenstandards anhand eines praktischen Beispiels 7. Ausblick				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Praktikum mit Testat Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens, N.N.				

12	Sonstige Informationen Dustdar, S.; Gall, H.; Hauswirth, M.: Software-Architekturen für verteilte Systeme, Springer-verlag, Berlin Heidelberg 2003 Andresen, A.: Komponentenbasierte Software-Entwicklung (Kapitel 6-10), Carl Hanser Verlag München Wien 2004 Emmerich.W.: Konstruktion von verteilten Objekten, dpunkt-Verlag Heidelberg, 2003
-----------	--

Wahlfach aus Liste 1 „Methoden der Informatik“ Spezielle Methoden der Programmierung					
Kenn-Nr. 5.3	Work-load 150 h	Credits 5 cps	Studien-semester 5. /6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Teilnehmer erhalten einen Überblick über verschiedene Programmierparadigmen und Sprachkonzepte sowie ihre Anwendungsgebiete. Sie lernen ausgewählte Konzepte an entsprechenden Programmiersprachen genauer kennen und verstehen Aufbau und Prinzipien von Programmiersprachen. Die Teilnehmer werden befähigt, selbständig neue Sprachkonstrukte zu erlernen und einzuordnen. Nach erfolgreichem Abschluss des Moduls sind sie in der Lage, problem- und lösungsorientiert ein geeignetes Paradigma und eine entsprechende Programmiersprache auszuwählen und einzusetzen.				
3	Inhalte <ul style="list-style-type: none"> • Überblick über Programmierparadigmen/-sprachen • Vertiefung ausgewählter Paradigmen <ul style="list-style-type: none"> ◦ funktionale Programmierung (beispielsweise Closure) ◦ nebenläufige/parallele Programmierung: Threads, Aktoren (am Beispiel von Scala) ◦ Domainspezifische Sprachen (intern, extern, Beispiele) • Werkzeuge für Programmiersprachen <ul style="list-style-type: none"> ◦ Einstieg Compilerbau (Grammatiken, Scanner, Parser, Compilergeneratoren (beispielsweise ANTLR)) • Domainspezifische Sprachen, XText 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Praktikum mit Testat Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips				
12	Sonstige Informationen <ul style="list-style-type: none"> • A. Grama, A. Gupta, G. Karypis, V. Kumar: Introduction to Parallel Computing, Addison-Wesley, 2003 • M. Erwig: Grundlagen funktionaler Programmierung, Oldenbourg, 1999 • R. Sethi: Programming Languages: Concepts and Constructs, Addison-Wesley 2003 				

Wahlfach aus Liste 1 "Methoden der Informatik"

Mustererkennung & Bildverarbeitung

Kenn-Nr. 5.4	Workload 150 h	Credits 5 cps	Studien- semester 5. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die wesentlichen Methoden der Mustererkennung und Bildverarbeitung. Insbesondere verstehen die Studierenden das Zusammenspiel von Bildverarbeitung und Klassifikation zur Extraktion von Informationen aus digitalen Bildern, um beispielsweise Problemstellungen in den Bereichen Computer Vision und Medizinische Bildverarbeitung lösen zu können. Die Studierenden erlernen im Praktikum sowohl das Erstellen von eigenen Programmen zur Bildverarbeitung als auch das Arbeiten mit einer Bildverarbeitungssoftware.				
3	Inhalte Überblick und Anwendungsgebiete Bildaufnahme Bildrepräsentation und -kompression Bildverbesserung und Filter Segmentierung Registrierung Merkmalsextraktion und -reduktion Klassifikationsverfahren Clustering				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Praktikum mit Testat Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: K. D. Tönnies, Grundlagen der Bildverarbeitung, Pearson Studium, 2005 R. C. Gonzales, R. E. Woods, Digital Image Processing, 3rd. Ed., Prentice Hall International, 2008 G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008				

Wahlfach aus Liste 1 „Methoden der Informatik“ Ausgewählte Konzepte von Datenbanksystemen					
Kenn-Nr. 5.5	Work-load 150 h	Credits 5 cps	Studien-semester 6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Datenbanksysteme sind hochkomplexe Softwaresysteme, zu deren sicherer Anwendung und Nutzung ein vertieftes Wissen der Implementierung und Administration gehört. Die Studierenden erwerben theoretische Kenntnisse in fortgeschrittenen Datenbanktechniken. Sie fällen begründete Entscheidungen für die Anwendung dieser Techniken. Sie können Datenbanken installieren und administrieren und dabei die Anforderungen der Anwendungssoftware analysieren und umsetzen. Sie können die Performanz von Datenbanken analysieren und durch geeignete technische Maßnahmen steigern. Sie können fortgeschrittene SQL formulieren und sinnvoll anwenden. Sie können SQL zur Performanzsteigerung analysieren und optimieren (SQL-Tuning).				
3	Inhalte <ul style="list-style-type: none"> • Architektur von Datenbanksystemen • Verwaltung des Hintergrundspeichers • Pufferverwaltung • Dateiorganisation und Zugriffsstrukturen • Spezielle Indexstrukturen • Basisalgorithmen für Datenbankoperationen • Fortgeschrittene SQL • Optimierung von Anfragen • Transaktionsmodelle • Transaktionsverwaltung • Wiederherstellung und Datensicherung 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Praktikum mit Testat Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking				
12	Sonstige Informationen Literatur: Saake, G., Sattler, K.-U., Datenbanken: Implementierungstechniken, Heidelberg 2011 Härder, Th., Rahm, E., Datenbanksysteme: Konzepte und Techniken der Implementierung, Berlin 2001 Aktuelle Literatur zu neueren Datenbanktechnologien				

Wahlfach aus Liste 1 „Methoden der Informatik“ Künstliche Intelligenz					
Kenn-Nr. 5.6	Work-load 150 h	Credits 5 cps	Studien-semester 5. /6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Methoden der Künstlichen Intelligenz finden zunehmend breiteren Einsatz bei Assistenz- und Entscheidungsunterstützungssystemen, bei Optimierungsproblemen, beim automatisierten Problemlösungen sowie in der Automatisierung.</p> <p>Die Veranstaltung führt in die Prinzipien und Methoden und Verfahren der modernen Künstlichen Intelligenz und deren Anwendung in intelligenten Systemen ein.</p> <p>Insbesondere werden die grundlegenden Methoden im Bereich der Künstlichen Intelligenz wie Konzepte des Problemlösens, der Wissensrepräsentation, der logischen Programmierung, des maschinellen Lernens und von neuronalen Netzen sowie des Planens vermittelt.</p> <p>Mit der erfolgreichen Teilnahme an der Vorlesung und den Übungen werden Kompetenzen zur Auswahl und Anwendung von Methoden der Künstlichen Intelligenz für eine konkrete Problemstellung erworben. Die Teilnehmer sind in der Lage, die erlernten Methoden auf andere Bereiche und Problemstellungen zu übertragen und anzuwenden.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Problemlösen: <ul style="list-style-type: none"> ○ Suchstrategien ○ Genetische/Evolutionäre Algorithmen ○ Einführung Constraintsolving ○ Spiele (Minimax Algorithmus, alpha-beta-Pruning) • Wissensrepräsentation, Prädikatenlogik, Inferenz • Logische Programmierung • Maschinelles Lernen: <ul style="list-style-type: none"> ○ Entscheidungsbaumverfahren ○ Neuronale Netze ○ Unüberwachtes Lernen • Autonome mobile Systeme • Wechselnde Inhalte der Praktika zu aktuellen Themen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Theoretische Informatik, Programmiermethoden				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Praktikum mit Testat Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				

10	Stellenwert der Note für die Endnote 5/210
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips
12	Sonstige Informationen <ul style="list-style-type: none">• Russel, Stuart/Norvig, Peter: "Artificial Intelligence. A Modern Approach", Prentice Hall, 2010.• Chris Bishop: "Pattern Recognition and Machine Learning", Springer, 2007• S.Haykin: "Neural Networks and Learning Machines: A Comprehensive Foundation", Prentice Hall, 2008

Wahlfach aus Liste 1 "Methoden der Informatik"					
Verlässliche Systeme/Security					
Kenn-Nr.	Workload	Credits	Studien-semester	Häufigkeit des Angebots	Dauer
5.7	150 h	5 cps	5. Sem.	Bei Nachfrage alle 2 Semester	1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden verstehen die Zusammenhänge und Unterschiede von Safety (Verlässlichkeit) und Security (Sicherheit) von Systemen und Software. Sie kennen sowohl Techniken zur Bewertung von Verlässlichkeit und Sicherheit von Software sowie zur Konstruktion von verlässlicher und sicherer Software. Anhand von Übungen erlernen die Studierenden, wie die Sicherheit eines Systems verbessert wird, sowohl hinsichtlich deren Ausfallsicherheit als auch deren Schutz vor Angriffen von außen.				
3	Inhalte Terminologie Vertraulichkeit, Integrität, Verfügbarkeit, Fehlertoleranz Kryptographie, Authentifizierung, Protokolle Gefahren- und Risikoanalyse sowie Sicherheitsplan Vorgehensweisen für Entwicklung von sicherer Software und Systeme				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Praktikum mit Testat Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König, N.N.				
12	Sonstige Informationen Literatur: N. Storey, Safety Critical Computer Systems, Addison Wesley, 1996 C. P. Pfleeger, Security in Computing, 4th ed., Prentice Hall 2007				

Wahlfach aus Liste 1 „Methoden der Informatik“ Methoden der Computergrafik					
Kenn-Nr. 5.8	Work-load 150 h	Credits 5 cps	Studien-semester 5. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS/60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Aufbauend auf den erworbenen Grundlagen der Vorlesung Computergrafik erlangen die Studierenden vertiefende Kenntnisse in den Bereichen Algorithmen und Datenstrukturen der Computergrafik, geometrische Modellierung, Visualisierung und Human Computer Interaction. In praktischen Beispielen werden die behandelnden Verfahren eingeübt.				
3	Inhalte <ul style="list-style-type: none"> • Algorithmen und Datenstrukturen: Verfahren aus der algorithmische Geometrie. • Einführung in die geometrische Modellierung: Bezier, B-Spline Kurven und Flächen. • Globale Beleuchtungsmethoden: Raytracing, Radiosity. • Virtual Reality Systeme. • Human Computer Interaction: Wahrnehmungspsychologie, Informationsverarbeitung. 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Praktikum mit Testat Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller				
12	Sonstige Informationen <ul style="list-style-type: none"> • Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition. • Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics – Principles and Practice, Addison-Wesley 				

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Anwendungen der Computergrafik					
Kenn-Nr. 5.9	Work-load 450 h	Credits 15 cps	Studien-semester 5./6. Sem.	Häufigkeit des Angebots bei Nachfrage alle zwei Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Aufbauend auf den erworbenen Grundlagen der Vorlesung Computergrafik bearbeiten die Studierenden aktuelle Problemstellungen aus der Computergrafik und sind in der Lage geeignete Methoden zu evaluieren und aktuelle Werkzeuge aus der Computergrafik auszuwählen. Die zu erstellenden Softwareapplikationen kommen aus den Anwendungsfeldern Mustererkennung & Bildverarbeitung, algorithmische Geometrie, geometrische Modellierung, Animation, physikalische Simulation, Virtual Reality, Informationsvisualisierung und Human Computer Interaction. Optional können auch interdisziplinäre Projekte bearbeitet werden.				
3	Inhalte <ul style="list-style-type: none"> • Projektarbeiten im Team zu computergrafisch spezifischen Anwendungen aus aktuellen Forschungsgebieten und Forschungskooperationen mit Industriepartnern. • Visualisierungstechniken: Methoden zur optimierten Darstellung. • Effiziente Datenstrukturen der Computergrafik, • Geometrische Modellierung: Bezier und B-Spline Techniken. • Human Computer Interaction und effektive Informationsverarbeitung. 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Erfolgreicher Abschluss der Projektarbeit				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 15/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller				
12	Sonstige Informationen <ul style="list-style-type: none"> • Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition. • Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics – Principles and Practice, Addison-Wesley 				

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Software Qualität					
Kenn-Nr. 5.10	Work-load 450 h	Credits 15 cps	Studien-semester 5./6. Sem.	Häufigkeit des Angebots bei Nachfrage alle zwei Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen Softwaresysteme werden zunehmend komplexer und übernehmen immer wichtigere Aufgaben im Alltag. Umso größer wird der Bedarf an der Vermittlung von Techniken und Methoden zum systematischen Test von Software. Nach dem Besuch der Veranstaltung haben die Studierenden folgende Kompetenzen erworben:</p> <ul style="list-style-type: none"> • Verständnis für die zunehmende Bedeutung von systematischem Softwaretest und die damit verbundenen Problemstellungen • Kenntnis vom allgemeinen Testprozess • Kenntnis von verschiedenen Prüf- und Testverfahren und ihrer praktischen Anwendung • Auswahl und Bewertung von unterschiedlichen Verfahren und Tools zur Erstellung von Testfällen • Kenntnis und Vorgehen von/bei verschiedenen Teststufen (Unit-, Komponenten-, Integrations-, System-, Abnahme- und Wartungstest) sowie Anwendung in der Praxis • Anwendung von statischen Prüfungen für Dokumente der Softwareentwicklung • Kenntnisse der Aufgaben des Testmanagements • Fähigkeiten, durch die vermittelten Techniken Software von höherer Qualität zu entwickeln <p>Die Veranstaltung dient der Vermittlung grundlegender theoretischer Kenntnisse und Fähigkeiten, wobei die praktische Anwendung im Projekt erlernt und vertieft werden soll. Durch die teamorientierte Projektarbeit werden die Projektmanagement- und Selbst-Kompetenzen der Teilnehmer vertieft. Nach der erfolgreichen Teilnahme an diesem Modul ist eine Prüfung zum "Certified Tester – Foundation Level" nach ISTQB bei einem vom German Testing Board zertifizierten Prüfungsinstitut möglich.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Grundlagen des Softwaretestens, Testprinzipien, fundamentaler Testprozess • Testen im Software-Life-Cycle, Teststufen und -arten • Statische Testtechniken • Dynamische Testtechniken, Testdesign <ul style="list-style-type: none"> ○ Spezifikationsbasiert (Blackbox) ○ Strukturbasiert (Whitebox) • Testmanagement • Werkzeuge • Wechselnde Inhalte der Projekte zu aktuellen Themen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				

7	Prüfungsformen Projektarbeit
8	Voraussetzungen für die Vergabe von Kreditpunkten Erfolgreicher Abschluss der Projektarbeit
9	Verwendung des Moduls (in anderen Studiengängen) -
10	Stellenwert der Note für die Endnote 15/210
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips
12	Sonstige Informationen <ul style="list-style-type: none">• Graham et.al.: "Foundations of Software Testing", Thomson Learning, 2008• Spillner, T. Linz: "Basiswissen Softwaretest", dpunkt-Verlag, 2010• G.J. Myers: "The Art of Software Testing", John Wiley, 2011

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Webengineering					
Kenn-Nr. 5.11	Work-load 450 h	Credits 15 cps	Studien-semester 5. /6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Verständnis von aktuellen Konzepten, Methoden, Techniken, Werkzeugen und Erfahrungen zur ingenieurmäßigen Entwicklung von Web-Anwendungen sowie ihre praktische Anwendung in der eigenen Projektarbeit in Web-Entwicklerteams. <ul style="list-style-type: none"> • Bewerten von potentiellen Risiken von Web-Anwendungen • Befähigung, zukünftige Entwicklungen im Bereich des Web-Engineering zu verfolgen und zu beurteilen Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei. Teamorientierte Projektarbeit erhöht ferner Projektmanagement- und Selbst-Kompetenzen.				
3	Inhalte Einführung in Web-Engineering (Motivation, Definition, Grundprinzipien) <ul style="list-style-type: none"> • Produktentwicklung • Requirements Engineering für Web-Anwendungen • Modellierung von Web-Anwendungen • Architektur von Web-Anwendungen • Testen von Web-Anwendungen • Web-Projektmanagement • Qualitätsaspekte (Usability, Performanz, Sicherheit) • Semantische Web-Anwendungen • Web-Frameworks • Mobile Applikationen • Mobile Endgeräte 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Erfolgreicher Abschluss der Projektarbeit				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 15/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens				
12	Sonstige Informationen <ul style="list-style-type: none"> • Bernd Müller: "Java Server Faces", Hanser 2006 • Kappel, Pröll, Reich, Teschitzegger: Web-Engineering, dpunkt 2004 • Spiering Haiges: "HTML5-Apps" Franzis 2011 				

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Mobile Applikationen					
Kenn-Nr. 5.12	Work-load 450 h	Credits 15 cps	Studien-semester 5./6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Softwareentwicklung für mobile Geräte ist die Weiterentwicklung der traditionellen (Anwendungs-) Softwareentwicklung hin zu einer offenen, verteilten und dynamischen Technologie, die mobile, leistungsfähige Geräte mit traditionellen Strukturen des Internets und verteilten Software-Systemen verbindet.</p> <p>Ziel dieses Moduls ist es, dass die Studierenden</p> <ul style="list-style-type: none"> • die Besonderheiten und Randbedingungen sowie Konzepte und Lösungsansätze der mobilen Applikationsentwicklung verstehen und auch quantitativ abschätzen können • darauf zugeschnittene Lösungen der Systemarchitektur und der Anwendungsentwicklung kennen lernen • und diese Kenntnisse zur Lösung von konkreten Aufgabenstellungen praktisch anwenden können. <p>Die Studierenden erhalten einen Überblick über aktuelle Hardwareplattformen und mobile Betriebssysteme und lernen ein aktuelles System (beispielsweise Android) zu beherrschen. Nach dem Besuch der Veranstaltung sind sie in der Lage, unter Berücksichtigung von Usability-, Energie- und Sicherheitsaspekten sowohl native als auch Webapplikationen für dieses System zu erstellen, die internen Sensoren und Hardwarebausteine zu nutzen.</p> <p>Zur Vertiefung werden größere Projekte erarbeitet, beispielsweise Applikationen zur Steuerung mobiler Roboter oder mobile Anwendungen aus den Bereichen Computer Grafik oder Augmented Reality. Die teamorientierte Projektarbeit vertieft die Projektmanagement- und Selbst-Kompetenzen.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Technologie mobiler Anwendungen, Hardware, Betriebssysteme • Anwendungen & Anwendungsentwicklung <ul style="list-style-type: none"> ○ Aktuelle Software-Frameworks für mobile Anwendungen ○ Ressourcenmanagement in mobilen Systemen ○ Sicherheit für mobile Systeme und Anwendungen ○ Energieaspekte • Nutzung von Hardwarebausteinen mobiler Geräte <ul style="list-style-type: none"> ○ Einbindung vorhandener Sensoren und Schnittstellen ○ Ansteuern externer Hardware, beispielsweise Arduino • Wechselnde Inhalte der Projekte zu aktuellen Themen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Erfolgreicher Abschluss der Projektarbeit				

9	Verwendung des Moduls (in anderen Studiengängen) -
10	Stellenwert der Note für die Endnote 15/210
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips
12	Sonstige Informationen <ul style="list-style-type: none">• J. Roth: "Mobile Computing - Grundlagen, Technik, Konzepte", dpunkt-Verlag, 2005• Mike Bach: "Mobile Anwendungen mit Android", Addison-Wesley, 2011• Tom Igoe: "Making Things Talk, 2nd Edition", O'Reilly, 2012• R. Siegwart, I. R. Nourbakhsh: "Introduction to Autonomous Mobile Robots", mitp, 2011

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Softwareengineering für Embedded Systems					
Kenn-Nr. 5.13	Work-load 450 h	Credits 15 cps	Studien-semester 5./6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontakt-zeit 6 SWS/90 h	Selbst-studium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden wissen die Methoden des Softwareengineering für Embedded Systems anzuwenden unter Berücksichtigung der Randbedingungen dieser Systeme. Sie besitzen die Fähigkeit, die Entwicklung von Software für Embedded Systems von der Planung bis zur Realisierung durchzuführen. Die Veranstaltung baut auf den Kenntnissen der Lehrveranstaltung „Embedded Systems“ auf.				
3	Inhalte <ul style="list-style-type: none"> • Anforderungen / Requirements • Software Architektur und Design Patterns • Modellbasierte Entwicklung • UML/SysML für Embedded Systems • Embedded Betriebssysteme • Distributed Embedded Systems • Besonderheiten der Programmierung von Embedded Systems • Qualitätssicherung Als vertiefendes Projekt wird beispielsweise unter einem Betriebssystem für eingebettete Systeme (z.B. Embedded Linux, Android, RTOS) eine Anwendung entwickelt (z.B. für Computer Vision, Robotik, Mess- und Regelungstechnik).				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Erfolgreicher Abschluss der Projektarbeit				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 15/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen				

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Datenbankanwendungen					
Kenn-Nr. 5.14	Work-load 450 h	Credits 15 cps	Studien-semester 5. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Datenbanken sind die Grundlagen jedes großen Softwaresystems. Das Zusammenwirken von Programmierung und Datenbank ist dabei von größter Wichtigkeit. Die Studierenden kennen die Anforderungen verschiedener Softwaresystemklassen an Datenbanken. Sie können spezielle Datenmodelle für unterschiedliche Systemklassen entwerfen und implementieren. Sie entwerfen und realisieren ein Softwaresystem bestehend aus Anwendungsprogramm und darauf abgestimmter Datenbank. Sie können Programmlogik in der Datenbank mit Persistent Stored Modules implementieren. Die Teilnehmer können verschiedene APIs zur Anbindung von Anwenderprogrammen an Datenbanken verwenden.				
3	Inhalte <ul style="list-style-type: none"> • Softwaresysteme und Anforderungen an Datenbanken • Anforderungsanalyse für Datenbankanwendungen • Objektorientierte und ER-Modellierung • Persistent Stored Modules (SQL/PSM) • Trigger • Impedance Mismatch • Cursors • Objektrelationale Mappings – Object Language Bindings (SQL/OLB) • APIs und Frameworks für Datenbankzugriffe 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking				
12	Sonstige Informationen				

Wahlfach aus Liste 2 „Anwendungen der Informatik“ Anwendungen der künstlichen Intelligenz					
Kenn-Nr. 5.15	Work-load 450 h	Credits 15 cps	Studien-semester 5./6. Sem.	Häufigkeit des Angebots Bei Nachfrage alle zwei Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesung 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS/90 h	Selbststudium 360 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studenten sind in der Lage, geeignete Suchstrategien zur Lösungsermittlung auszuwählen und in Teams erfolgreich eine intelligente Applikation zu entwickeln. Mobile Geräte mit verschiedenen Sensoren wie Mobiltelefone und Mobile autonome Roboter werden erfolgreich in die Applikationen über Serviceschnittstellen integriert.				
3	Inhalte <ul style="list-style-type: none"> • Grundlagen der Modernen KI • Ansatz des intelligenten Agenten • Verschiedene Suchstrategien theoretisch und beispielhaft • Implementierung ausgesuchter Suchstrategien in passenden Programmiersprachen • Aktuelle Forschungs- und Anwendungsprojekte in Teamarbeit • eventbasierte Serviceschnittstellen • Middleware • Elemente der Robotik • autonome Service Roboter • Sensoren (Video, Audio) • Elemente der Bildverarbeitung, Sprachverarbeitung und Mustererkennung 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Erfolgreicher Abschluss der Projektarbeit				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 15/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens				
12	Sonstige Informationen Karsten Berns: "Autonomous Land Vehicles", Vieweg+Teubner 2009 Russel Norvig: "Künstliche Intelligenz", Pearson 2004				

IT-Recht					
Kenn-Nr. 6.1	Work-load 150 h	Credits 5 cps	Studien-semester 6. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS	Kontakt- unterricht 4 SWS/60 h	Selbst- studium 90 h	geplante Gruppengröße Seminaristischer Unterricht 35	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die wichtigsten rechtlichen Aspekte, mit denen sie während einer Tätigkeit im Gebiet der Informatik in Berührung kommen können. Insbesondere können sie einschätzen, welche Rechte und Pflichten bei Verträgen bezüglich Herstellung, Vertrieb und Nutzung von (Software/Hardware-) Produkten entstehen, mit welchen Schutzrechten geistiges Eigentum geschützt werden kann, wie Datenschutz beachtet werden muss, sowie mit welchen Folgen bei Rechtsverstößen zu rechnen ist.				
3	Inhalte Die Informationstechnologie betreffenden Inhalte sind querschnittlich im Wesentlichen aus: <ul style="list-style-type: none"> • Zivilrecht und Vertragsrecht • Produkthaftung • Datenschutz • Strafrecht • Telekommunikations-, Telemedien- und Internetrecht • Gewerblicher Rechtsschutz (u.a. Urheber-, Patent-, Markenrecht) 				
4	Lehrformen Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Hausarbeit, Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Seminaristischen Unterricht mit Testat. Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Matthias König, N.N.				
12	Sonstige Informationen Literatur: Schneider, Computerrecht – IT- und Computerrecht, 9. Aufl., DTV, 2010 Dreier, Vogel, Software- und Computerrecht, UTB, 2008				

Fachseminar					
Kenn-Nr. 6.2	Work-load 150 h	Credits 5 cps	Studien-semester 6.Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS	Kontakt- unterricht 4 SWS/60 h	Selbst- studium 90 h	geplante Gruppengröße Seminaristischer Unterricht 35	
2	Lernergebnisse (learning outcomes) / Kompetenzen Befähigung zum selbständigen wissenschaftlichen Arbeiten mit der Erschließung von Inhalten sowie deren verständliche Darstellung von fachlichen Themen sind für den beruflichen Alltag unabdingbar. Absolventen des Moduls <ul style="list-style-type: none"> • beherrschen selbstständiges Erarbeiten eines Fachthemas anhand von Fachliteratur und anderen Quellen • können ein Fachthema verständlich darstellen und präsentieren • erwerben kommunikative Kompetenz • vertiefen informatische Fachkompetenzen auf dem ausgewählten Themengebiet des Seminars 				
3	Inhalte Selbstorganisation und selbstständiges Erarbeiten eines Fachthemas <ul style="list-style-type: none"> • Wissensmanagement und Literaturstudium (Recherche, Umgang mit Zitaten und Zitieren von Fachliteratur) • Fachliches Schreiben für die schriftliche Ausarbeitung • Präsentationstechnik und Rhetorik für die Präsentation des Themas • Diskussion im Rahmen der Seminarteilnehmer und betreuenden Dozenten 				
4	Lehrformen Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Keine				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
7	Prüfungsformen Hausarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Teilnahme am Seminaristischen Unterricht mit Testat. Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Grit Behrens, N.N.				
12	Sonstige Informationen Publikationen zum gewählten Themengebiet in deutscher und englischer Sprache				

Praxisphase					
Kenn-Nr. 7.1	Work-load 450 h	Credits 18 cps	Studien-semester 7. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Tätigkeit bei individueller Praxisstelle	Kontaktzeit 450 h	Selbststudium Erfolgt am Arbeitsplatz	geplante Gruppengröße 1	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Praxisphase bietet den Studierenden die Möglichkeit, ihre in mehreren Semestern an der Hochschule erworbenen fachlichen Fähigkeiten in der Praxis zu erproben und zusätzlich wichtige Kompetenzen im außerfachlichen Bereich zu erwerben. Es spielt daher im Rahmen einer praxisorientierten und arbeitsmarktgerechten Ausbildung sowie zur Persönlichkeitsbildung eine zentrale Rolle. Die Lernergebnisse umfassen: <ul style="list-style-type: none"> • Orientierung im angestrebten Berufsfeld • Erwerb praktischer Kenntnisse und Kennenlernen berufstypischer Arbeitsweisen • Kennenlernen technischer und organisatorischer Zusammenhänge, die für das Berufsfeld typisch sind. • Beteiligung am Arbeitsprozess entsprechend dem Ausbildungsstand • Praktische Ausbildung an fest umrissenen, konkreten Projekten 				
3	Inhalte Praktische Tätigkeit mit deutlichem Informatik-Schwerpunkt, z.B. <ul style="list-style-type: none"> • Durchführung von Anforderungsanalysen, Erstellung Softwareentwurf • Programmierung • Datenbankentwurf und -implementierung • Realisierung von Web-Anwendungen • Netzwerkplanung, Sicherheitsanalysen • Verarbeitung von Graphikdaten, Visualisierung Rahmenbedingungen: <ul style="list-style-type: none"> • Kürzere tägliche Arbeitszeiten als ein halber Tag sind nicht erlaubt. • Das gesamte Praktikum ist innerhalb von 9 Monaten abzuleisten. • Krankheits- und andere Ausfallzeiten zählen dabei nicht mit. • Das Berufspraktikum muss bei einer Praktikumsstelle absolviert werden. • Über das Berufspraktikum ist ein Bericht von 20 Seiten anzufertigen. • Das Praktikum wird von einem hochschulseitigen Betreuer überwacht. 				
4	Lehrformen Arbeitstätigkeit				
5	Teilnahmevoraussetzungen Alle Modulprüfungen aus Semester 1 bis 4 bestanden (siehe auch § 24 BPO):				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO. Fristgerechte Abgabe des Praxisberichtes.				
7	Prüfungsformen Bewerteter Praxisbericht als Testat				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 18/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Dozenten im Studiengang Informatik (Becking, Behrens, George, Gips, Kreienkamp, König, Müller, Hoffmann, N.N.)				
12	Sonstige Informationen Arbeitsmaterialien und Literatur entsprechen der individuellen Aufgabenstellung				

Bachelorarbeit					
Kenn-Nr. 7.2	Work-load 360 h	Credits 12 cps	Studien-semester 7. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen 0,3 SWS Individuelle dozentengebundene Betreuung	Kontakt-zeit 10 h	Selbst-studium 350 h	geplante Gruppengröße 1	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die selbständige Lösung eines praxisbezogenen Themas nach wissenschaftlichen Grundsätzen gehört zu den beruflichen Fähigkeiten eines Informatikers. Die systematische Bearbeitung und praxisbezogene Umsetzung einer Aufgabenstellung sowie die zusammenhängende Darstellung von Berichten und Publikationen dient der Kommunikation zwischen Fachleuten und stellt sicher, dass erworbene Kenntnisse und Fähigkeiten erhalten bleiben. Die Studierenden lernen, wie eine Aufgabe methodisch bearbeitet und in einem vorgegebenen Zeitrahmen mit einem klar strukturierten Ergebnis dargestellt wird, indem sie sich schnell in eine neue Aufgabenstellung einarbeiten und ihr Wissen in einem speziellen Gebiet selbstständig vertiefen. Die Studierenden lernen übliche Werkzeuge und Methoden zur Arbeitsunterstützung einzusetzen, wenden eine Reihe fachspezifischer Fähigkeiten, Fertigkeiten und Techniken an, um Aufgaben selbständig zu lösen, diese zu analysieren, zu bewerten und in einem Gesamtkontext darzustellen.				
3	Inhalte Mit der Bachelorarbeit soll unter Beweis gestellt werden, dass Studierende in der Lage sind, eine komplexe Fragestellung mit wissenschaftlichen Methoden innerhalb eines begrenzten Zeitraums zu lösen und das dabei erworbene theoretische und praktische Wissen nachvollziehbar zu dokumentieren. 1. Konkretisieren der Aufgabenstellung 2. Erstellung eines Zeitplans 3. Evaluation und Aufstellung der zu verwendenden Techniken und Methoden 4. Erstellung eines Software-Konzeptes 5. Implementierung und Dokumentation der Software-Lösung 6. Gesamtbetrachtung, Test und Bewertung der Lösung 7. Darstellung der Lösung in Form der Bachelor-Arbeit				
4	Lehrformen Individuelle dozentengebundene Betreuung Selbständige Erstellung der Bachelorarbeit				
5	Teilnahmevoraussetzungen Bestandene Modulprüfungen gem. Studienverlaufsplan bis auf vier Modulprüfungen (siehe auch BPO §29). Kenntnisse in der Breite des studierten Faches				
6	Prüfungsvoraussetzungen Es gilt § 17 „Fortschrittsregelung“ dieser BPO. Fristgemäße Abgabe der Bachelorarbeit im Prüfungsamt (BPO §31)				
7	Prüfungsformen Von zwei Prüfern bewertete Bachelorarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Bachelorarbeit				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 12/210				
11	Modulbeauftragte/r und hauptamtlich Lehrende Dozenten im Studiengang Informatik (Becking, Behrens, George, Gips, Kreienkamp, König, Müller, Hoffmann, N.N.)				
12	Sonstige Informationen Arbeitsmaterialien und Literatur entsprechen der individuellen Aufgabenstellung				