

**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)**

<b>1. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
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	-
<b>Summen</b>	<b>30</b>	<b>22</b>	<b>11</b>	<b>-</b>	<b>7</b>	<b>4</b>
<b>2. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
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	-	-
<b>Summen</b>	<b>30</b>	<b>23</b>	<b>9</b>	<b>4</b>	<b>5</b>	<b>5</b>
<b>3. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
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	-	-
<b>Summen</b>	<b>30</b>	<b>22</b>	<b>9</b>	<b>4</b>	<b>-</b>	<b>9</b>
<b>4. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
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	-	-
<b>Summen</b>	<b>30</b>	<b>24</b>	<b>8</b>	<b>4</b>	<b>-</b>	<b>12</b>
<b>5. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
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
<b>Summen</b>	<b>30</b>	<b>18</b>	<b>8</b>	<b>-</b>	<b>2</b>	<b>8</b>
<b>6. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
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
<b>Summen</b>	<b>30</b>	<b>18</b>	<b>4</b>	<b>8</b>	<b>-</b>	<b>6</b>
<b>7. Semester</b>	<b>cps</b>	<b>SWS</b>	<b>V</b>	<b>SU</b>	<b>Ü</b>	<b>P</b>
Praxisphase	18	-	-	-	-	-
Bachelorarbeit	12	-	-	-	-	-
<b>Summen</b>	<b>30</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>
<b>Gesamtsummen</b>	<b>210</b>	<b>122</b>	<b>49</b>	<b>20</b>	<b>14</b>	<b>38</b>

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

<b>Einführung in das Berufsfeld</b>					
<b>Kenn-Nr.</b> 1.1	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 1. Sem.	<b>Häufigkeit des Angebots</b> Jedes WS	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Übung 2 SWS	<b>Kontaktzeit</b> 4 SWS /60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Übung 30	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Geschichte der Informatik; Errungenschaften der Informatik</li> <li>• Informatik und Gesellschaft</li> <li>• Teilgebiete der Informatik</li> <li>• Berufsbilder von Informatikern</li> <li>• Branchen und Bereiche für Absolventen der Informatik</li> <li>• Persönliche Entwicklungsperspektiven für Absolventen der Informatik</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Seminar, Exkursionen				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Klausur				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Grit Behrens				
<b>12</b>	<b>Sonstige Informationen</b> -				

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

<b>Objektorientierte Programmierung</b>					
<b>Kenn-Nr.</b>	<b>Work-load</b>	<b>Credits</b>	<b>Studien-semester</b>	<b>Häufigkeit des Angebots</b>	<b>Dauer</b>
<b>1.3</b>	210 h	7 cps	1. Sem.	Jedes WS	1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 150 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Einführung in objektorientiertes Design</li> <li>• Einführung des Klassen- und Objektbegriffs</li> <li>• Nutzung einer Entwicklungsumgebung und eines Debuggers</li> <li>• Datentypen und Kontrollstrukturen</li> <li>• Einführung in die Vererbung, Polymorphie</li> <li>• Einführung in die Ausnahmebehandlung</li> <li>• Einführung von Collection-Klassen</li> <li>• Einführung in Swing</li> <li>• Ansätze zum Testen</li> <li>• Ansätze zum Dokumentieren (beispielsweise JavaDoc)</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum: die lt. Studienverlaufsplan 2 SWS Praktika werden durch 1 SWS tutorbetreutes Praktikum vor- bzw. nachbereitet				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistung der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 7/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Carsten Gips, N.N.				
<b>12</b>	<b>Sonstige Informationen</b> <ul style="list-style-type: none"> <li>• Krüger, G.: "Handbuch der Java-Programmierung", Addison-Wesley, 2011</li> <li>• Ullenboom, C.: "Java ist auch eine Insel", Galileo Computing, 2011</li> </ul>				

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

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

<b>Mathematik für Informatiker II</b>					
<b>Kenn-Nr.</b> 2.1	<b>Work-load</b> 240 h	<b>Credits</b> 8 cps	<b>Studien-semester</b> 2. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 3 SWS Übung 3 SWS	<b>Kontaktzeit</b> 6 SWS/90 h	<b>Selbststudium</b> 150 h	<b>geplante Gruppengröße</b> Vorlesung 60 Übung 30	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <b>Analysis II</b> <ul style="list-style-type: none"> <li>• Lokale und globale Approximation</li> <li>• Differentialgleichungen</li> <li>• Reellwertige Funktionen mit mehrerer Variablen</li> <li>• Differential- und Integralrechnung für Funktionen mehrerer Variablen</li> </ul> <b>Numerik</b> <ul style="list-style-type: none"> <li>• Fehler und Fehlerfortpflanzung</li> <li>• Elementare numerische Verfahren</li> <li>• Optimierung</li> </ul> <b>Diskrete Wahrscheinlichkeitsrechnung und Statistik</b> <ul style="list-style-type: none"> <li>• Klassische Wahrscheinlichkeit</li> <li>• Bedingte Wahrscheinlichkeit</li> <li>• Zufallsgrößen</li> <li>• Verteilungen</li> <li>• Statistik</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Übung				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Klausur				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 8/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Kerstin Müller, Dipl. Inf. Birgit Christina George				
<b>12</b>	<b>Sonstige Informationen</b> <b>Literatur</b> Papula: Mathematik für Ingenieure Bronstein, Semendjajev: Taschenbuch der Mathematik				

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

<b>Algorithmen und Datenstrukturen</b>					
<b>Kenn-Nr.</b> 2.3	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 2. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS / 60h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Begrifflichkeiten (Algorithmus, Komplexität)</li> <li>• Gleichwertige Lösungen               <ul style="list-style-type: none"> <li>○ Maximale Teilsumme</li> <li>○ Komplexitätsformel</li> <li>○ Datenstrukturen (Reihungen, verkettete Listen)</li> </ul> </li> <li>• Rekursion und Wiederholung (Rekursive Algorithmen, Abarbeitung von Datenstrukturen)</li> <li>• Suchen in               <ul style="list-style-type: none"> <li>○ Texten</li> <li>○ Sammlungen</li> <li>○ Reihungen</li> <li>○ verketteten Listen</li> <li>○ Hashtabellen</li> </ul> </li> <li>• Sortierverfahren (quadratische, unterquadratische, rekursive, logarithmische)</li> <li>• Baumstrukturen (Suchen, Darstellen, Sortieren, Durchwandern, Operationen)</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistungen der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Dipl.-Inf. B.C. George				
<b>12</b>	<b>Sonstige Informationen</b> Literaturangaben <ul style="list-style-type: none"> <li>• Andreas Solymosi und Ulrich Grude: „Grundkurs Algorithmen und Datenstrukturen in Java“, Vieweg+Teubner, Wiesbaden 2008</li> <li>• Ralf Hartmut Güting, Stefan Dieker: „Datenstrukturen und Algorithmen“, Vieweg +Teubner, Wiesbaden 2004</li> </ul>				

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

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

<b>Software Engineering</b>					
<b>Kenn-Nr.</b> 3.1	<b>Work-load</b> 210 h	<b>Credits</b> 7 cps	<b>Studien-semester</b> 3. Sem.	<b>Häufigkeit des Angebots</b> Jedes WS	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 150 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Einführung in Softwareengineering</li> <li>• Prozessmodellierung (Unternehmensprozesse, Aktivitätsdiagramme, Risikomanagement)</li> <li>• Vorgehensmodelle ( Wasserfallmodell, Prototypische Entwicklung, Iterative Entwicklung, Iterativ-inkrementelle Entwicklung, Allgemeines V-Modell, Agile Entwicklung wie Scrum und Extreme Programming)</li> <li>• Anforderungsanalyse (Stakeholder, Ziele, Uses Cases, Ableitung funktionaler Anforderungen, nicht funktionale Anforderungen, Lasten- und Pflichtenheft)</li> <li>• Grobdesign (Systemarchitektur, Ableitung der grundlegenden Klassen, Methoden, Sequenzdiagramm, Überlegungen zur Oberflächenentwicklung)</li> <li>• Programmgenerierung( CASE-Tools, Übersetzung von Klassen und Assoziationen, Arten der Objektzugehörigkeit, Software-Architektur)</li> <li>• Feindesign (Details im Kleinen, Model View Controller, GoF-Pattern)</li> <li>• Implementierungen (Verteilte Systeme, XML, Programmbibliotheken, Komponenten, Frameworks, Persistente Datenhaltung, Model Driven Architecture)</li> <li>• Oberflächengestaltung ( Useability, Ergonomie, Prüfung)</li> <li>• SW-Qualitätssicherung (Zusicherungen, Unit-Tests, Testverfahren, Testarten, Metriken)</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistungen der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 7/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Grit Behrens, N.N.				
<b>12</b>	<b>Sonstige Informationen</b> 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				

<b>Verteilte Systeme und Kommunikationsnetze</b>					
<b>Kenn-Nr.</b>	<b>Work-load</b>	<b>Credits</b>	<b>Studien-semester</b>	<b>Häufigkeit des Angebots</b>	<b>Dauer</b>
<b>3.2</b>	150 h	5 cps	3. Sem.	Jedes WS	1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen</b></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>				
<b>3</b>	<p><b>Inhalte</b></p> <ul style="list-style-type: none"> <li>• Grundlagen der Kommunikationsnetze <ul style="list-style-type: none"> <li>- Kommunikationsmodelle/-protokolle und Standards</li> <li>- Vermittlungsprinzipien</li> </ul> </li> <li>• Technologien für lokale Netze (LAN) <ul style="list-style-type: none"> <li>- Übertragungsmedien</li> <li>- Stochastische und deterministische Medienzugriffsverfahren</li> <li>- Ethernet-Technologien und Protokolle</li> </ul> </li> <li>• Protokolle der TCP/IP-Protokollfamilie (IP, ICMP, UDP, TCP, Protokolle der Anwendungsschicht)</li> <li>• Übersicht über Technologien für Weitverkehrsnetze</li> <li>• Grundlagen IP-basierter Routing-Protokolle</li> <li>• Router und Routerkonfiguration</li> <li>• Routing-Protokolle</li> <li>• Funktionsorientierte Client/Server-Systeme mit Remote Procedure Calls (RPC)</li> <li>• Objektorientierte, verteilte Systeme (CORBA, Java RMI etc.)</li> <li>• Sicherheit von Verteilten Anwendungen</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine.				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine.				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistungen der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> 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.

<b>Datenbanken</b>					
<b>Kenn-Nr.</b> 3.3	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 4. Sem.	<b>Häufigkeit des Angebots</b> Jedes WS	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen</b></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>				
<b>3</b>	<p><b>Inhalte</b></p> <ul style="list-style-type: none"> <li>• Aufgaben und Architektur von Datenbanksystemen</li> <li>• Clients und Schnittstellen zu Datenbanksystemen</li> <li>• Grundlagen des relationalen Modells</li> <li>• E/R Modellierung, logische und physische Datenmodelle, SQL Datentypen, Implementierung in wichtigen DBMS</li> <li>• Constraints, Assertions, Integrität, Domänen, Datentypen</li> <li>• SQL:2011, insbesondere SQL-schema statements, SQL-data statements, SQL-data change statements, SQL-transaction statements und SQL-connection statements</li> <li>• Transaktionskonzepte, Nebenläufigkeit, Isolationslevel</li> <li>• Datenbankschnittstellen (JDBC, ODBC)</li> <li>• Grundlagen von Persistent Stored Modules, Programmierung von PSM, Trigger</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistungen der Performanzprüfung				

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

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

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

<b>Technical English</b>					
<b>Kenn-Nr.</b> 3.5	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 3. Sem.	<b>Häufigkeit des Angebots</b> Jedes WS	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Seminaristischer Unterricht 4 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Seminaristischer Unterricht 35	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Computer Hardware: Ein- und Ausgabemedien, Drucker, Speichermedien</li> <li>• Wesentliche und kreative Software: Betriebssystem, Textverarbeitung, Tabellenkalkulation und Datenbanken, Desktop-Publishing, Multimedia</li> <li>• Programmierung : Programmiersprachen, Java, Webgestaltung</li> <li>• Zukunftstechnologien: Kommunikationssysteme, Netzwerke, Videospiele</li> <li>• Internet: E-Mail, das Web, Videokonferenzführung</li> <li>• Beschäftigungsmöglichkeiten im Bereich IKT</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Seminaristischer Unterricht				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Keine				
<b>7</b>	<b>Prüfungsformen</b> Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Teilnahme am Seminaristischen Unterricht mit Testat Bestandene Klausur				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> Keine				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Cathrine Stones				
<b>12</b>	<b>Sonstige Informationen</b>				

**Softwareprojekt**

<b>Kenn-Nr.</b> 4.1	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 4. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Praktikum 4 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Einrichtung und Durchführung eines Projektes</li> <li>• Erstellung eines Pflichtenhefts auf Basis der Vorgaben eines fiktiven „Kunden“, Durchführung einer Aufwandsschätzung auf Basis dieses Pflichtenhefts</li> <li>• Aufstellen eines Projektplanes und Verfahren zur Projektverfolgung und des Risikomanagements</li> <li>• Implementierung im Team (Version-Management, Build-Verfahren, Abstimmungsprozesse, Schnittstellen)</li> <li>• Präsentation von Ergebnissen und Zwischenergebnissen</li> <li>• Einsatz aktueller Technologien zur Implementierung der Anwendung</li> <li>• Abschlusspräsentation</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Projektarbeit				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> Keine				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> N.N., Prof. Dr. Becking, Prof. Dr. Behrens, Prof. Dr. Gips, Prof. Dr. Hoffmann				
<b>12</b>	<b>Sonstige Informationen</b>				

<b>Betriebssysteme</b>					
<b>Kenn-Nr.</b> 4.2	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 4. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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)				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Einführung: Aufgaben, Arten und Strukturen von Betriebssystemen</li> <li>• Kurzvorstellung aktuelle Rechnerarchitekturen (Pipeline, Cache, Branch Prediction)</li> <li>• Nebenläufigkeit, Prozessverwaltung, Scheduling</li> <li>• Synchronisation und Kommunikation</li> <li>• Speicherverwaltung: Virtueller Speicher, Segmentierung, Paging, Seitenersetzungsalgorithmen</li> <li>• Ein-/Ausgabe: zeichen- und blockorientierte Geräte</li> <li>• Dateisysteme: Beispiele, Aufbau, Fehlertoleranz</li> <li>• Treiber-Modelle und -Programmierung</li> <li>• Sicherheit: Authentifikation, Schutzmechanismen, Autorisierung, vertrauenswürdige Systeme</li> <li>• Wechselnde Inhalte der Praktika zu aktuellen Themen (z.B. Bash-Skripting, Entwicklung eigener Shell, Programmierung Linux Kerneltreiber)</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistung der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Martin Hoffmann				
<b>12</b>	<b>Sonstige Informationen</b> <ul style="list-style-type: none"> <li>• Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, 2009</li> <li>• Stallings: "Operating Systems: Internals and Design Principles", Prentice Hall, International, 2011</li> </ul>				

**Embedded Systems**

<b>Kenn-Nr. 4.3</b>	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 4. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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).				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Überblick und Anwendungsgebiete</li> <li>• Besonderheiten und Anforderungen</li> <li>• Zusammenspiel Software und Hardware</li> <li>• Entwurfsschritte</li> <li>• Spezifikation und Modellierung</li> <li>• Realisierung und Implementierung</li> <li>• Validierung und Evaluation</li> </ul> Praktische Umsetzung eingebetteter Systeme an Anwendungsbeispielen.				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistungen der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Matthias König				
<b>12</b>	<b>Sonstige Informationen</b> Literatur: P. Marwedel, Eingebettete Systeme, Springer 2008				

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

**12****Sonstige Informationen**

Literaturangaben:

- 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

<b>Webbasierte Anwendungen</b>					
<b>Kenn-Nr.</b> 4.5	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 4. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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"> <li>• Kennen typischer Anwendungsgebiete Web-basierter Anwendungen</li> <li>• Fähigkeit zum problemadäquaten Entwurf und zur Realisierung Web-basierter Anwendungen, einschl. Beurteilung und Auswahl geeigneter Basistechnologien</li> </ul>				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Einführung, Klassifikation von Web-Anwendungen, Architekturen</li> <li>• Grundlagen (HTTP, Sessionmanagement, Standardisierung, W3C)</li> <li>• Sicherheit von Webanwendungen</li> <li>• Webseiten gestalten (XHTML, Cascading Stylesheets, HTML5)</li> <li>• Webuseability (Suchmaschinenoptimierung, Metriken auswerten, Besucherverhalten verstehen, Websites optimieren)</li> <li>• Clientseitige Technologien: JavaScript, Ajax, DOM</li> <li>• Serverseitige Technologien: PHP, Servlets, JSP's</li> <li>• Webservices: SOAP, WSDL</li> <li>• Mehrschichtenarchitekturen, Frameworks zu deren Umsetzung: JSF, Struts</li> <li>• Applikationsserver (Aufgaben, Dienste, Beispiele): Apache Tomcat</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Teilleistungen der Performanzprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Grit Behrens				
<b>12</b>	<b>Sonstige Informationen</b> 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				

<b>SW-Projektmanagement</b>					
<b>Kenn-Nr.</b> 4.6	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 4. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Seminaristischer Unterricht 4 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Seminaristischer Unterricht 35	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Projekte als Problemlösungsprozesse</li> <li>• Gründung, Organisation und Strukturierung von Projekten</li> <li>• Projektplanung</li> <li>• Projektsteuerung</li> <li>• Management von Software-Projekten</li> <li>• Werkzeuge im Projektmanagement</li> <li>• Kommunikation und Dokumentation als Querschnittsaufgabe</li> <li>• Qualitätssicherung</li> <li>• Projektmanagementsysteme</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Seminaristischer Unterricht				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Klausur, mündliche Prüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Teilnahme am Seminaristischen Unterricht mit Testat Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Dominic Becking				
<b>12</b>	<b>Sonstige Informationen</b>				

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

## Wahlfach aus Liste 1 "Methoden der Informatik"

### Komponentenbasierte SW-Entwicklung

<b>Kenn-Nr.</b> 5.2	<b>Workload</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 5. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> 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				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Mündliche Prüfung, Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Praktikum mit Testat Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Grit Behrens, N.N.				

<b>12</b>	<b>Sonstige Informationen</b> 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
-----------	--

<b>Wahlfach aus Liste 1 „Methoden der Informatik“ Spezielle Methoden der Programmierung</b>					
<b>Kenn-Nr.</b> 5.3	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 5. /6. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Überblick über Programmierparadigmen/-sprachen</li> <li>• Vertiefung ausgewählter Paradigmen               <ul style="list-style-type: none"> <li>◦ funktionale Programmierung (beispielsweise Closure)</li> <li>◦ nebenläufige/parallele Programmierung: Threads, Aktoren (am Beispiel von Scala)</li> <li>◦ Domainspezifische Sprachen (intern, extern, Beispiele)</li> </ul> </li> <li>• Werkzeuge für Programmiersprachen               <ul style="list-style-type: none"> <li>◦ Einstieg Compilerbau (Grammatiken, Scanner, Parser, Compilergeneratoren (beispielsweise ANTLR))</li> </ul> </li> <li>• Domainspezifische Sprachen, XText</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> mündliche Prüfung, Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Praktikum mit Testat Bestandene Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Carsten Gips				
<b>12</b>	<b>Sonstige Informationen</b> <ul style="list-style-type: none"> <li>• A. Grama, A. Gupta, G. Karypis, V. Kumar: Introduction to Parallel Computing, Addison-Wesley, 2003</li> <li>• M. Erwig: Grundlagen funktionaler Programmierung, Oldenbourg, 1999</li> <li>• R. Sethi: Programming Languages: Concepts and Constructs, Addison-Wesley 2003</li> </ul>				

## Wahlfach aus Liste 1 "Methoden der Informatik" Mustererkennung & Bildverarbeitung

<b>Kenn-Nr.</b> 5.4	<b>Workload</b> 150 h	<b>Credits</b> 5 cps	<b>Studien- semester</b> 5. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> Überblick und Anwendungsgebiete Bildaufnahme Bildrepräsentation und -kompression Bildverbesserung und Filter Segmentierung Registrierung Merkmalsextraktion und -reduktion Klassifikationsverfahren Clustering				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Mündliche Prüfung, Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Praktikum mit Testat Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Matthias König				
<b>12</b>	<b>Sonstige Informationen</b> 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

<b>Kenn-Nr.</b> 5.5	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 6. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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).				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Architektur von Datenbanksystemen</li> <li>• Verwaltung des Hintergrundspeichers</li> <li>• Pufferverwaltung</li> <li>• Dateioorganisation und Zugriffsstrukturen</li> <li>• Spezielle Indexstrukturen</li> <li>• Basisalgorithmen für Datenbankoperationen</li> <li>• Fortgeschrittene SQL</li> <li>• Optimierung von Anfragen</li> <li>• Transaktionsmodelle</li> <li>• Transaktionsverwaltung</li> <li>• Wiederherstellung und Datensicherung</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b>				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Mündliche Prüfung, Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Praktikum mit Testat Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Dominic Becking				
<b>12</b>	<b>Sonstige Informationen</b> 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				

<b>Wahlfach aus Liste 1 „Methoden der Informatik“ Künstliche Intelligenz</b>					
<b>Kenn-Nr.</b> 5.6	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 5. /6. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen</b></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>				
<b>3</b>	<p><b>Inhalte</b></p> <ul style="list-style-type: none"> <li>• Problemlösen: <ul style="list-style-type: none"> <li>○ Suchstrategien</li> <li>○ Genetische/Evolutionäre Algorithmen</li> <li>○ Einführung Constraintsolving</li> <li>○ Spiele (Minimax Algorithmus, alpha-beta-Pruning)</li> </ul> </li> <li>• Wissensrepräsentation, Prädikatenlogik, Inferenz</li> <li>• Logische Programmierung</li> <li>• Maschinelles Lernen: <ul style="list-style-type: none"> <li>○ Entscheidungsbaumverfahren</li> <li>○ Neuronale Netze</li> <li>○ Unüberwachtes Lernen</li> </ul> </li> <li>• Autonome mobile Systeme</li> <li>• Wechselnde Inhalte der Praktika zu aktuellen Themen</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Theoretische Informatik, Programmiermethoden				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Mündliche Prüfung, Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Teilnahme am Praktikum mit Testat Bestandene Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				

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

## Wahlfach aus Liste 1 "Methoden der Informatik" Verlässliche Systeme/Security

<b>Kenn-Nr.</b> 5.7	<b>Workload</b> 150 h	<b>Credits</b> 5 cps	<b>Studien- semester</b> 5. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 2 SWS	<b>Kontaktzeit</b> 4 SWS/60 h	<b>Selbststudium</b> 90 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> Terminologie Vertraulichkeit, Integrität, Verfügbarkeit, Fehlertoleranz Kryptographie, Authentifizierung, Protokolle Gefahren- und Risikoanalyse sowie Sicherheitsplan Vorgehensweisen für Entwicklung von sicherer Software und Systeme				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Mündliche Prüfung, Klausur				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Teilnahme am Praktikum mit Testat Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Matthias König, N.N.				
<b>12</b>	<b>Sonstige Informationen</b>  Literatur: N. Storey, Safety Critical Computer Systems, Addison Wesley, 1996 C. P. Pfleeger, Security in Computing, 4th ed., Prentice Hall 2007				

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

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

<b>Wahlfach aus Liste 2 „Anwendungen der Informatik“ Software Qualität</b>					
<b>Kenn-Nr.</b> 5.10	<b>Work-load</b> 450 h	<b>Credits</b> 15 cps	<b>Studien-semester</b> 5./6. Sem.	<b>Häufigkeit des Angebots</b> bei Nachfrage alle zwei Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 4 SWS	<b>Kontaktzeit</b> 6 SWS/90 h	<b>Selbststudium</b> 360 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen</b></p> <p>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.</p> <p>Nach dem Besuch der Veranstaltung haben die Studierenden folgende Kompetenzen erworben:</p> <ul style="list-style-type: none"> <li>• Verständnis für die zunehmende Bedeutung von systematischem Softwaretest und die damit verbundenen Problemstellungen</li> <li>• Kenntnis vom allgemeinen Testprozess</li> <li>• Kenntnis von verschiedenen Prüf- und Testverfahren und ihrer praktischen Anwendung</li> <li>• Auswahl und Bewertung von unterschiedlichen Verfahren und Tools zur Erstellung von Testfällen</li> <li>• Kenntnis und Vorgehen von/bei verschiedenen Teststufen (Unit-, Komponenten-, Integrations-, System-, Abnahme- und Wartungstest) sowie Anwendung in der Praxis</li> <li>• Anwendung von statischen Prüfungen für Dokumente der Softwareentwicklung</li> <li>• Kenntnisse der Aufgaben des Testmanagements</li> <li>• Fähigkeiten, durch die vermittelten Techniken Software von höherer Qualität zu entwickeln</li> </ul> <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.</p> <p>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>				
<b>3</b>	<p><b>Inhalte</b></p> <ul style="list-style-type: none"> <li>• Grundlagen des Softwaretestens, Testprinzipien, fundamentaler Testprozess</li> <li>• Testen im Software-Life-Cycle, Teststufen und -arten</li> <li>• Statische Testtechniken</li> <li>• Dynamische Testtechniken, Testdesign <ul style="list-style-type: none"> <li>◦ Spezifikationsbasiert (Blackbox)</li> <li>◦ Strukturbasiert (Whitebox)</li> </ul> </li> <li>• Testmanagement</li> <li>• Werkzeuge</li> <li>• Wechselnde Inhalte der Projekte zu aktuellen Themen</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				

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

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

<b>Wahlfach aus Liste 2 „Anwendungen der Informatik“ Mobile Applikationen</b>					
<b>Kenn-Nr.</b> 5.12	<b>Work-load</b> 450 h	<b>Credits</b> 15 cps	<b>Studien-semester</b> 5./6. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 4 SWS	<b>Kontaktzeit</b> 6 SWS/90 h	<b>Selbststudium</b> 360 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<p><b>Lernergebnisse (learning outcomes) / Kompetenzen</b></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"> <li>• die Besonderheiten und Randbedingungen sowie Konzepte und Lösungsansätze der mobilen Applikationsentwicklung verstehen und auch quantitativ abschätzen können</li> <li>• darauf zugeschnittene Lösungen der Systemarchitektur und der Anwendungsentwicklung kennen lernen</li> <li>• und diese Kenntnisse zur Lösung von konkreten Aufgabenstellungen praktisch anwenden können.</li> </ul> <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>				
<b>3</b>	<p><b>Inhalte</b></p> <ul style="list-style-type: none"> <li>• Technologie mobiler Anwendungen, Hardware, Betriebssysteme</li> <li>• Anwendungen &amp; Anwendungsentwicklung <ul style="list-style-type: none"> <li>◦ Aktuelle Software-Frameworks für mobile Anwendungen</li> <li>◦ Ressourcenmanagement in mobilen Systemen</li> <li>◦ Sicherheit für mobile Systeme und Anwendungen</li> <li>◦ Energieaspekte</li> </ul> </li> <li>• Nutzung von Hardwarebausteinen mobiler Geräte <ul style="list-style-type: none"> <li>◦ Einbindung vorhandener Sensoren und Schnittstellen</li> <li>◦ Ansteuern externer Hardware, beispielsweise Arduino</li> </ul> </li> <li>• Wechselnde Inhalte der Projekte zu aktuellen Themen</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Projektarbeit				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Erfolgreicher Abschluss der Projektarbeit				

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

<b>Wahlfach aus Liste 2 „Anwendungen der Informatik“ Softwareengineering für Embedded Systems</b>					
<b>Kenn-Nr.</b> 5.13	<b>Work-load</b> 450 h	<b>Credits</b> 15 cps	<b>Studien-semester</b> 5./6. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 4 SWS	<b>Kontakt-zeit</b> 6 SWS/90 h	<b>Selbst-studium</b> 360 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Anforderungen / Requirements</li> <li>• Software Architektur und Design Patterns</li> <li>• Modellbasierte Entwicklung</li> <li>• UML/SysML für Embedded Systems</li> <li>• Embedded Betriebssysteme</li> <li>• Distributed Embedded Systems</li> <li>• Besonderheiten der Programmierung von Embedded Systems</li> <li>• Qualitätssicherung</li> </ul> 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).				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Projektarbeit				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Erfolgreicher Abschluss der Projektarbeit				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 15/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Matthias König				
<b>12</b>	<b>Sonstige Informationen</b>				

<b>Wahlfach aus Liste 2 „Anwendungen der Informatik“ Datenbankanwendungen</b>					
<b>Kenn-Nr.</b> 5.14	<b>Work-load</b> 450 h	<b>Credits</b> 15 cps	<b>Studien-semester</b> 5. Sem.	<b>Häufigkeit des Angebots</b> Bei Nachfrage alle 2 Semester	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Vorlesung 2 SWS Praktikum 4 SWS	<b>Kontaktzeit</b> 6 SWS/90 h	<b>Selbststudium</b> 360 h	<b>geplante Gruppengröße</b> Vorlesung 60 Praktikum 15	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> <ul style="list-style-type: none"> <li>• Softwaresysteme und Anforderungen an Datenbanken</li> <li>• Anforderungsanalyse für Datenbankanwendungen</li> <li>• Objektorientierte und ER-Modellierung</li> <li>• Persistent Stored Modules (SQL/PSM)</li> <li>• Trigger</li> <li>• Impedance Mismatch</li> <li>• Cursors</li> <li>• Objektrelationale Mappings – Object Language Bindings (SQL/OLB)</li> <li>• APIs und Frameworks für Datenbankzugriffe</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Vorlesung, Praktikum				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Performanzprüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestehen der Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Dr. Dominic Becking				
<b>12</b>	<b>Sonstige Informationen</b>				

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

<b>IT-Recht</b>					
<b>Kenn-Nr.</b> 6.1	<b>Work-load</b> 150 h	<b>Credits</b> 5 cps	<b>Studien-semester</b> 6. Sem.	<b>Häufigkeit des Angebots</b> Jedes SoSe	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> Seminaristischer Unterricht 4 SWS	<b>Kontakt- unterricht</b> 4 SWS/60 h	<b>Selbst- studium</b> 90 h	<b>geplante Gruppengröße</b> Seminaristischer Unterricht 35	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> Die Informationstechnologie betreffenden Inhalte sind querschnittlich im Wesentlichen aus: <ul style="list-style-type: none"> <li>• Zivilrecht und Vertragsrecht</li> <li>• Produkthaftung</li> <li>• Datenschutz</li> <li>• Strafrecht</li> <li>• Telekommunikations-, Telemedien- und Internetrecht</li> <li>• Gewerblicher Rechtsschutz (u.a. Urheber-, Patent-, Markenrecht)</li> </ul>				
<b>4</b>	<b>Lehrformen</b> Seminaristischer Unterricht				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> keine				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO.				
<b>7</b>	<b>Prüfungsformen</b> Hausarbeit, Klausur, mündliche Prüfung				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Teilnahme am Seminaristischen Unterricht mit Testat. Bestandene Modulprüfung				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 5/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Prof. Matthias König, N.N.				
<b>12</b>	<b>Sonstige Informationen</b> Literatur: Schneider, Computerrecht – IT- und Computerrecht, 9. Aufl., DTV, 2010 Dreier, Vogel, Software- und Computerrecht, UTB, 2008				

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

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

<b>Bachelorarbeit</b>					
<b>Kenn-Nr.</b> 7.2	<b>Work-load</b> 360 h	<b>Credits</b> 12 cps	<b>Studien-semester</b> 7. Sem.	<b>Häufigkeit des Angebots</b> Jedes WS	<b>Dauer</b> 1 Semester
<b>1</b>	<b>Lehrveranstaltungen</b> 0,3 SWS Individuelle dozentengebundene Betreuung	<b>Kontakt-zeit</b> 10 h	<b>Selbst-studium</b> 350 h	<b>geplante Gruppengröße</b> 1	
<b>2</b>	<b>Lernergebnisse (learning outcomes) / Kompetenzen</b> 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.				
<b>3</b>	<b>Inhalte</b> 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				
<b>4</b>	<b>Lehrformen</b> Individuelle dozentengebundene Betreuung Selbständige Erstellung der Bachelorarbeit				
<b>5</b>	<b>Teilnahmevoraussetzungen</b> Bestandene Modulprüfungen gem. Studienverlaufsplan bis auf vier Modulprüfungen (siehe auch BPO §29). Kenntnisse in der Breite des studierten Faches				
<b>6</b>	<b>Prüfungsvoraussetzungen</b> Es gilt § 17 „Fortschrittsregelung“ dieser BPO. Fristgemäße Abgabe der Bachelorarbeit im Prüfungsamt (BPO §31)				
<b>7</b>	<b>Prüfungsformen</b> Von zwei Prüfern bewertete Bachelorarbeit				
<b>8</b>	<b>Voraussetzungen für die Vergabe von Kreditpunkten</b> Bestandene Bachelorarbeit				
<b>9</b>	<b>Verwendung des Moduls (in anderen Studiengängen)</b> -				
<b>10</b>	<b>Stellenwert der Note für die Endnote</b> 12/210				
<b>11</b>	<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Dozenten im Studiengang Informatik (Becking, Behrens, George, Gips, Kreienkamp, König, Müller, Hoffmann, N.N.)				
<b>12</b>	<b>Sonstige Informationen</b> Arbeitsmaterialien und Literatur entsprechen der individuellen Aufgabenstellung				