

**Modulhandbuch**  
**Lehramt-Studiengang**  
**„Informatik“**

**Ruprecht-Karls-Universität Heidelberg**  
**Fakultät für Mathematik und Informatik**

**Fassung vom 18.07.2017 zur Prüfungsordnung vom 22.07.2010**  
**mit letzter Änderung vom 11.11.2011**

**Studienform:** Vollzeit

**Art des Studiengangs:** Grundständig

**Regelstudienzeit:** 10 Semester

**Anzahl der im Studiengang zu erwerbenden Leistungspunkte:** 104

**Studienstandort:** Heidelberg

**Anzahl der Studienplätze:** Keine Zulassungsbeschränkung

**Gebühren/Beiträge:** Gemäß allgemeiner Regelung der Universität Heidelberg

# Präambel

## Einordnung und Gesamtdarstellung des Lehramt-Studiengangs Informatik

Anknüpfend an ihr Leitbild und ihre Grundordnung verfolgt die Universität Heidelberg in ihren Studiengängen fachliche, fachübergreifende und berufsfeldbezogene Ziele in der umfassenden akademischen Bildung und für eine spätere berufliche Tätigkeit ihrer Studierenden. Das daraus folgende Kompetenzprofil wird als für alle Disziplinen Qualifikationsprofil in den Modulhandbüchern aufgenommen und in den spezifischen Qualifikationszielen sowie dem Curriculum und Modulen des Lehramts-Studiengangs Informatik umgesetzt:

- Entwicklung von fachlichen Kompetenzen mit ausgeprägter Forschungsorientierung;
- Entwicklung transdisziplinärer Dialogkompetenz;
- Aufbau von praxisorientierter Problemlösungskompetenz;
- Entwicklung von personalen und Sozialkompetenzen;
- Förderung der Bereitschaft zur Wahrnehmung gesellschaftlicher Verantwortung auf der Grundlage der erworbenen Kompetenzen.

Das Studium für das Lehramt an Gymnasien hat das Ziel, die Professionalität und Qualität künftiger Lehrkräfte am Gymnasium zu sichern. AbsolventInnen haben die fachwissenschaftlichen, fachdidaktischen, bildungswissenschaftlichen und ethisch-philosophischen Kenntnisse und Kompetenzen, einschließlich personaler Kompetenzen, die für die Erziehungs- und Bildungsarbeit an Gymnasien und Gemeinschaftsschulen sowie für die Übernahme in den Vorbereitungsdienst erforderlich sind.

Die AbsolventInnen des Lehramt-Studienganges Informatik

- verfügen über Kenntnisse der Praktischen, Theoretischen, Technischen und Angewandten Informatik sowie der Methoden der Mathematik,
- können eine umfangreiche informatische Aufgabe eigenverantwortlich planen, durchführen, dokumentieren und präsentieren,
- kennen die verschiedenen Sicht- und Arbeitsweisen der Informatik von ingenieurmäßigen Zugängen wie Analysieren und Konstruieren über mathematische Verfahren zur Erkenntnisgewinnung wie Formalisieren und Beweisen bis hin zu gesellschaftswissenschaftlichen und empirischen Methoden wie Experimentieren und Simulieren,
- können Realsituationen analysieren und strukturieren, um diese der Verarbeitung mit Methoden der Informatik zugänglich zu machen,

- können informatikspezifische Inhalts- und Prozesskonzepte auf andere Anwendungsfelder übertragen und ihre erworbenen informatischen Kompetenzen in außerinformatischen Kontexten nutzen,
- können fachlich anspruchsvolle Sachverhalte adäquat mündlich und schriftlich darstellen und sich selbstständig informatische Inhalte aneignen,
- können informatische Sachverhalte in verschiedenen Anwendungsbezügen und Sachzusammenhängen sowie gesellschaftliche Auswirkungen erfassen, bewerten und erklären,
- können Informatik als Disziplin charakterisieren und die Funktion und das Bild der Informatik beziehungsweise der informatischen Bildung in der Gesellschaft reflektieren,
- können den zentralen Beitrag des Informatikunterrichts zur Allgemeinbildung in der Informationsgesellschaft erkennen und sind dadurch in der Lage, geeignete Inhalte und Schwerpunkte zu wählen und didaktisch reduziert für den Schulunterricht aufzubereiten,
- kennen geeignete didaktische Umgebungen, die der Erarbeitung der Inhalte dienen und sind in der Lage, sich neue Werkzeuge selbstständig anzueignen sowie diese hinsichtlich ihrer Eignung für den gymnasialen Unterricht zu bewerten,
- können die Langlebigkeit und Übertragbarkeit zentraler informatischer Fachkonzepte beurteilen und aktuelle Entwicklungen fachlich bewerten, bezüglich ihrer schulischen Relevanz auswählen und aufbereiten,
- können aktuelle Entwicklungstendenzen zur Schulinformatik reflektieren und vertreten eine kritische Offenheit bezüglich neuer Entwicklungen der Informatik.

# Inhaltsverzeichnis

<b>1</b>	<b>Studienverlaufspläne</b>	<b>5</b>
<b>2</b>	<b>Pflichtbereich</b>	<b>10</b>
2.1	Pflichtmodule Informatik	10
	Einführung in die Praktische Informatik	11
	Programmierkurs	13
	Programmierkurs mit Python	14
	Algorithmen und Datenstrukturen	16
	Betriebssysteme und Netzwerke	18
	Einführung in die Technische Informatik	20
	Einführung in die Theoretische Informatik	22
	Datenbanken 1	24
	Einführung in Software Engineering	26
	Seminar	28
	Fortgeschrittenenpraktikum	29
2.2	Pflichtmodule Mathematik	30
	Mathematik für Informatiker 1	31
2.3	Pflichtmodule Fachdidaktik	32
	Fachdidaktik 1	33
	Fachdidaktik 2	35
<b>3</b>	<b>Wahlpflichtbereich</b>	<b>37</b>

# 1 Studienverlaufspläne

Es wird unterschieden zwischen mehreren Studienverlaufsplänen abhängig vom gewählten zweiten Hauptfach. Diese Pläne stellen jeweils nur eine Möglichkeit für den Studienverlauf dar und sind insbesondere nicht bindend, sie müssen vielmehr angepasst werden auch mit Blick auf das zweite Hauptfach.

## Studienverlaufsplan mit weiterem Hauptfach Mathematik

<b>1. Semester:</b>	Einführung in die Praktische Informatik	8 LP
	Programmierkurs	3 LP
<b>2. Semester:</b>	Algorithmen und Datenstrukturen	8 LP
	Betriebssysteme und Netzwerke	8 LP
<b>3. Semester:</b>	Einführung in die Technische Informatik	8 LP
	Fachdidaktik 1 (ein Teil)	2 LP
<b>4. Semester:</b>	Einführung in die Theoretische Informatik	8 LP
	Fachdidaktik 1 (anderer Teil)	2 LP
	<b><i>Zwischenprüfung</i></b>	4 LP
<b>5. Semester:</b>	<i>Schulpraxissemester</i>	
<b>6. Semester:</b>	Datenbanken 1	8 LP
	Fachdidaktik 2	6 LP
<b>7. Semester:</b>	Einführung in Software Engineering	8 LP
	Wahlpflicht	
<b>8. Semester:</b>	Seminar	4 LP
	Fortgeschrittenenpraktikum	8 LP
<b>9. oder 10. Semester:</b>	Wahlpflicht	
	Wissenschaftliche Arbeit (falls Informatik 1. Hauptfach)	
	<b><i>Staatsexamensprüfung</i></b>	

Das Modul *Fachdidaktik 1* besteht aus zwei Teilen, die unabhängig voneinander absolviert werden können.

Insgesamt sind 19 LP Wahlpflicht zu belegen.

## Studienverlaufsplan mit weiterem Hauptfach Bildende Kunst oder Musik

<b>1. Semester:</b>	Einführung in die Praktische Informatik	8 LP
	Mathematik	8 LP
<b>2. Semester:</b>	Algorithmen und Datenstrukturen	8 LP
	Betriebssysteme und Netzwerke	8 LP
<b>3. Semester:</b>	Einführung in die Technische Informatik	8 LP
	Programmierkurs	3 LP
	Fachdidaktik 1 (ein Teil)	2 LP
<b>4. Semester:</b>	Einführung in die Theoretische Informatik	8 LP
	Fachdidaktik 1 (anderer Teil)	2 LP
	<i>Zwischenprüfung</i>	4 LP
<b>5. Semester:</b>	<i>Schulpraxissemester</i>	
<b>6. Semester:</b>	Datenbanken 1	8 LP
	Fachdidaktik 2	6 LP
<b>7. Semester:</b>	Einführung in Software Engineering	8 LP
	Wahlpflicht	
<b>8. Semester:</b>	Seminar	4 LP
	Fortgeschrittenenpraktikum	8 LP
<b>9. oder 10. Semester:</b>	Wahlpflicht	
	Wissenschaftliche Arbeit (falls Informatik 1. Hauptfach)	
	<i>Staatsexamensprüfung</i>	

Das Modul *Fachdidaktik 1* besteht aus zwei Teilen, die unabhängig voneinander absolviert werden können.

Insgesamt sind 14 LP Wahlpflicht zu belegen.

## Studienverlaufsplan mit weiterem Hauptfach Gesundheit und Gesellschaft (Care)

<b>1. Semester:</b>	Einführung in die Praktische Informatik	8 LP
	Mathematik	8 LP
<b>2. Semester:</b>	Algorithmen und Datenstrukturen	8 LP
	Betriebssysteme und Netzwerke	8 LP
<b>3. Semester:</b>	Einführung in die Technische Informatik	8 LP
	Fachdidaktik 1 (ein Teil)	2 LP
<b>4. Semester:</b>	Einführung in die Theoretische Informatik	8 LP
	Fachdidaktik 1 (anderer Teil)	2 LP
	<b><i>Zwischenprüfung</i></b>	4 LP
<b>5. Semester:</b>	<i>Schulpraxissemester</i>	
<b>6. Semester:</b>	Datenbanken 1	8 LP
	Fachdidaktik 2	6 LP
<b>7. Semester:</b>	Einführung in Software Engineering	8 LP
<b>8. Semester:</b>	Seminar	4 LP
	Fortgeschrittenenpraktikum	8 LP
<b>9. oder 10. Semester:</b>	Wissenschaftliche Arbeit (falls Informatik 1. Hauptfach)	
	<b><i>Staatsexamensprüfung</i></b>	

Das Modul *Fachdidaktik 1* besteht aus zwei Teilen, die unabhängig voneinander absolviert werden können.



## Studienverlaufsplan ohne weiteres Hauptfach Mathematik, Bildende Kunst/Musik oder Care

<b>1. Semester:</b>	Einführung in die Praktische Informatik	8 LP
	Mathematik	8 LP
<b>2. Semester:</b>	Algorithmen und Datenstrukturen	8 LP
	Betriebssysteme und Netzwerke	8 LP
<b>3. Semester:</b>	Einführung in die Technische Informatik	8 LP
	Programmierkurs	3 LP
	Fachdidaktik 1 (ein Teil)	2 LP
<b>4. Semester:</b>	Einführung in die Theoretische Informatik	8 LP
	Fachdidaktik 1 (anderer Teil)	2 LP
	<b><i>Zwischenprüfung</i></b>	4 LP
<b>5. Semester:</b>	<i>Schulpraxissemester</i>	
<b>6. Semester:</b>	Datenbanken 1	8 LP
	Fachdidaktik 2	6 LP
<b>7. Semester:</b>	Einführung in Software Engineering	8 LP
	Wahlpflicht	
<b>8. Semester:</b>	Seminar	4 LP
	Fortgeschrittenenpraktikum	8 LP
<b>9. oder 10. Semester:</b>	Wahlpflicht	
	Wissenschaftliche Arbeit (falls Informatik 1. Hauptfach)	
	<b><i>Staatsexamensprüfung</i></b>	

Das Modul *Fachdidaktik 1* besteht aus zwei Teilen, die unabhängig voneinander absolviert werden können.

Insgesamt sind 11 LP Wahlpflicht zu belegen.

## 2 Pflichtbereich

Im Folgenden sind die Pflichtmodule des Lehramt-Studiengangs Informatik beschrieben, unterteilt in die Bereiche Informatik, Mathematik und Fachdidaktik.

### 2.1 Pflichtmodule Informatik

Nachfolgend sind die Pflichtmodule im Bereich Informatik für den Lehramt-Studiengang Informatik beschrieben.

Beim Programmierkurs gibt es zwei Varianten den Programmierkurs (IPK), welcher die Programmiersprache C++ behandelt, und den Programmierkurs mit Python (IPK-Python), welcher die Programmiersprache Python behandelt. Von diesen beiden Varianten ist nur eine zu absolvieren, je nach dem aktuellen Lehrangebot.

Bei der Kombination mit dem weiteren Hauptfach Gesundheit und Gesellschaft (Care) entfällt der Programmierkurs.

## Einführung in die Praktische Informatik

<b>Code</b> IPI	<b>Name</b> Einführung in die Praktische Informatik	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, B.Sc. Mathematik
<b>Lernziele</b>	Kenntnis der unten angegebenen Inhalte Fähigkeit, kleine Programme in C++ zu entwerfen, zu realisieren, zu testen und Eigenschaften der Programme zu ermitteln. Umgang mit einfachen Programmierwerkzeugen.	
<b>Inhalt</b>	<p>Die Lehrveranstaltung führt in die Entwicklung von Software im Kleinen ein.</p> <p>Überblick über die Praktische Informatik.</p> <p>Technische und formale Grundlagen der Programmierung.</p> <p>Sprachliche Grundzüge (Syntax und Semantik von Programmiersprachen).</p> <p>Einführung in die Programmierung (Wert, elementare Datentypen, Funktion, Bezeichnerbindung, Sichtbarkeit von Bindungen, Variable, Zustand, Algorithmus, Kontrollstrukturen, Anweisung, Prozedur)</p> <p>Weitere Grundelemente der Programmierung (Typisierung, Parametrisierung, Rekursion, strukturierte Datentypen, insbesondere z.B. Felder, Listen, Bäume).</p> <p>Grundelemente der objektorientierten Programmierung (Objekt, Referenz, Klasse, Vererbung, Subtypbildung).</p> <p>Abstraktion und Spezialisierung (insbesondere Funktions-, Prozedurabstraktion, Abstraktion und Spezialisierung von Klassen) .</p> <p>Spezifikation und Verifikation von Algorithmen, insbesondere einfache Testtechniken.</p> <p>Terminierung.</p> <p>Einfache Komplexitätsanalysen.</p> <p>Einfache Algorithmen (Sortierung).</p>	
<b>Voraussetzungen</b>	keine	

<b>Pruefungs- modalitaeten</b>	<p>Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung.</p> <p>Es wird am Ende der Vorlesungszeit eine Klausur angeboten.</p> <p>Wird diese nicht bestanden so kann die Prüfungsleistung in einer zweiten Klausur vor Beginn der nächsten Vorlesungszeit erbracht werden.</p>
<b>Nuetzliche Literatur</b>	<p>Wird jährlich aktualisiert</p>

## Programmierkurs

<b>Code</b> IPK	<b>Name</b> Programmierkurs	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Praktikum 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 30 h praktische Übung am Rechner 30 h Hausaufgaben	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	Die Studierenden können selbstständig Programme und Lösungen von Programmieraufgaben in C++ entwerfen, realisieren und testen sind in der Lage mit gängigen Programmierwerkzeugen und Tools unter Linux umzugehen	
<b>Inhalt</b>	Die Lehrveranstaltung vertieft die Programmierkenntnisse aus dem Modul Einführung in die Praktische Informatik (IPI). Im Vordergrund steht der Erwerb praktischer Fähigkeiten. Die Studierenden lernen algorithmische Lösungen systematisch in Programme umzusetzen. Es wird die Programmiersprache C++ unter dem Betriebssystem Linux verwendet. Behandelt werden neben einer Einführung in Linux Datentypen, Deklarationen, Variablen, Schleifen, Kontrollstrukturen, Blockstrukturen, Prozeduren und Funktion, Zeiger, Konzepte der objektorientierten Programmierung (Klassen, Methoden und Templates). Es werden weiterhin die Tätigkeiten der Neuentwicklung, des Testens und der Fehlersuche sowie die Bewertung von Ergebnissen erlernt.	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung. Im Wintersemester wird am Ende der Vorlesungszeit eine Klausur angeboten. Wird diese nicht bestanden so kann die Prüfungsleistung in einer zweiten Klausur vor Beginn der nächsten Vorlesungszeit erbracht werden. Im Sommersemester wird nur eine Klausur angeboten.	
<b>Nützliche Literatur</b>		

## Programmierkurs mit Python

<b>Code</b> IPK-Python	<b>Name</b> Programmierkurs mit Python	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Praktikum 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 30 h praktische Übung am Rechner 30 h Hausaufgaben	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	Die Teilnehmer können selbständig Programme und Lösungen von Programmieraufgaben in Python entwerfen, realisieren und testen. Sie kennen die gängigen Bibliotheken und Programmierwerkzeuge für die Entwicklung von Python-Software.	
<b>Inhalt</b>	<p>Die Lehrveranstaltung vertieft die Programmierkenntnisse aus dem Modul Einführung in die Praktische Informatik (IPI). Im Vordergrund steht der Erwerb praktischer Fähigkeiten. Die Studierenden lernen, algorithmische Lösungen systematisch in Programme umzusetzen.</p> <p>Es wird die Programmiersprache Python verwendet. Der Kurs erklärt zunächst grundlegende Sprachbestandteile wie Datentypen, Ausdrücke, Kontrollstrukturen und Schleifen und behandelt danach ausführlich die wichtigsten Module der Standardbibliothek, die exemplarisch für die Bibliotheken moderner Programmiersprachen stehen.</p> <p>Neben der prozeduralen Programmierung erlernen die Teilnehmer die Konzepte der objektorientierten und der funktionalen Programmierung (Klassen und Methoden bzw. Iteratoren, Generatoren und Lambdas) als wichtige alternative Paradigmen.</p> <p>Der Kurs geht weiterhin auf Python als Sprache für das wissenschaftliche Rechnen ein und behandelt grundlegende Bibliotheken für numerische Berechnungen (insbesondere numpy) sowie für die Optimierung der Laufzeit (insbesondere cython).</p> <p>Außerdem machen sich die Teilnehmer mit der Programmentwicklung in einer Entwicklungsumgebung, dem systematischen Testen sowie der Fehlersuche vertraut.</p>	
<b>Voraussetzungen</b>	keine	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung. Im Wintersemester wird am Ende der Vorlesungszeit eine Klausur angeboten. Wird diese nicht bestanden so kann die Prüfungsleistung in einer zweiten Klausur vor Beginn der nächsten Vorlesungszeit erbracht werden. Im Sommersemester wird nur eine Klausur angeboten.
<b>Nuetzliche Literatur</b>	

## Algorithmen und Datenstrukturen

<b>Code</b> IAD	<b>Name</b> Algorithmen und Datenstrukturen	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	Die Studierenden sind mit den wichtigsten Datenstrukturen der Informatik vertraut, kennen die Methoden zur Analyse der Laufzeit von Algorithmen, sind mit den Basisproblemen Sortieren und Suchen vertraut und kennen die abhängig von der konkreten Anwendung besten Algorithmen, kennen Datenstrukturen für Graphen und können elementare Probleme auf Graphen lösen, haben die Methoden zur Suche von Textmustern gelernt, sind in der Lage, den Schwierigkeitsgrad von Problemen zu beurteilen	
<b>Inhalt</b>	Grundlagen zu Algorithmen (Eigenschaften, Darstellungsmöglichkeiten) Analyse der Laufzeit von Algorithmen (Lösen von Rekursionsgleichungen, amortisierte Komplexität) Grundlegende Datenstrukturen (Liste, Stack, Queue) Sortierverfahren (Insertion-, Selection-, Quick-, Heap-, Merge-Sort, Sortieren ohne Schlüsselvergleiche) Manipulation von Mengen (Prioritätswarteschlangen, Systeme von disjunkten Mengen) Suchen (Medianproblem, lineare Listen, Suchbäume) Hash-Verfahren (Hashing mit Verkettung, offenes Hashing, Analyse von Kollisionen) Einfache Graphenalgorithmen (Speicherung von Graphen, Breitensuche, Tiefensuche, aufspannende Bäume, kürzeste Wege) Suchen in Texten (Suche nach Wörtern und Mustern, Tries) Komplexität (Turing-Maschinen, Klassen P und NP)	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)	



<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
<b>Nuetzliche Literatur</b>	z. B.: Sedgewick, R.: Algorithmen, Pearson, 2002 Cormen, T.H., Leiserson, Ch.E., Rivest, R.L.: Introduction to Algorithms, MIT press, 2001 Kleinberg J., Tardos, E.: Algorithm Design, 2005

## Betriebssysteme und Netzwerke

<b>Code</b> IBN	<b>Name</b> Betriebssysteme und Netzwerke	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 4 SWS, Gruppen-Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	Die Veranstaltung führt in die Grundlagen der Betriebssysteme und Netzwerke moderner Rechner ein. Sie vermittelt notwendiges Grundwissen über die Abläufe innerhalb eines Rechners und die Abwicklung der Kommunikation zwischen ihnen.	
<b>Inhalt</b>	Themen der Betriebssystemtechnik sind: * Prozesse und ihre Verwaltung * Verwaltung des Speichers im Rechner * Prozesssynchronisation * Nebenläufigkeit und Verklemmungen * Scheduling * Eingabe/Ausgabe und Dateiverwaltung * Themen der Netzwerktechnik sind: * Schichtenmodell der Rechnerkommunikation * Direktverbindungsnetze * Paketvermittlung * Internetworking * Ende-zu-Ende-Protokolle * Überlastkontrolle * Anwendungen	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI)	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen und erfolgreiche Teilnahme an einer schriftlichen oder mündlichen Abschlussprüfung je nach Anzahl der TeilnehmerInnen	

<b>Nuetzliche Literatur</b>	* Moderne Betriebssysteme. Andrew S. Tanenbaum und David J. Wetherall, 5. (oder frühere) Auflage, Pearson Studium, August 2012. * Operating system concepts. Abraham Silberschatz, Peter Baer Galvin, und Greg Gagne. 9. (oder frühere) Auflage, John Wiley & Sons, Dezember 2012. * Computernetzwerke: der Top-Down-Ansatz. James F. Kurose und Keith W. Ross. 6. (oder frühere Auflage , Pearson Studium, März 2014.
---------------------------------	--

## Einführung in die Technische Informatik

<b>Code</b> ITE	<b>Name</b> Einführung in die Technische Informatik	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	Die Studierenden erwerben Kenntnisse über den grundsätzlichen Aufbau und der Funktionsweise von Rechnersystemen: Möglichkeiten und Grenzen der Hardware Verständnis für spezifisches Systemverhalten Entwicklung hardwarenaher Programme (Programmierung in Maschinensprache und Treiberentwicklung) Darstellung und Verarbeitung von Information in Rechnern	
<b>Inhalt</b>	Schaltalgebra Digitale Schaltungen Sequentielle Logik Technologische Grundlagen Programmierbare Logikbausteine Zahlendarstellung und Codierung Rechnerarithmetik Ein einfacher Prozessor Pipelineverarbeitung von Befehlen Vorhersage von Sprüngen Peripherie	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung. Es wird am Ende der Vorlesungszeit eine Klausur angeboten. Wird diese nicht bestanden so kann die Prüfungsleistung in einer zweiten Klausur vor Beginn der nächsten Vorlesungszeit erbracht werden.	

<p><b>Nuetzliche Literatur</b></p>	<p>Standardwerke:  W. Schiffmann, R. Schmitz: Technische Informatik 2: Grundlagen der Computertechnik , Springer-Lehrbuch, Springer (2005)  Alan Clements: The Principles of Computer Hardware. 3rd Ed., Oxford Univ. Press, 2000.  Andrew S. Tanenbaum: Computerarchitektur. 5. Auflage, Pearson Studium, 2006  Ergänzungsliteratur:  Walter Oberschelp, Gottfried Vossen: Rechneraufbau und Rechnerstrukturen.10.Aufl., Oldenbourg, 2006.  John D. Carpinelli: Computer Systems, Organization &amp; Architecture.Addison-Wesley, 2001.</p>
--	--

## Einführung in die Theoretische Informatik

<b>Code</b> ITH	<b>Name</b> Einführung in die Theoretische Informatik	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 4 SWS, Gruppen-Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	<p>Die Studierenden sind mit grundlegenden Aspekten des Berechenbarkeitsbegriffs vertraut, insbesondere mit dessen anschaulicher Bedeutung und den Formalisierungen durch Turingmaschinen, Registermaschinen und rekursive Funktionen, kennen den Beweis der Äquivalenz der verschiedenen Formalisierungen des Berechenbarkeitsbegriffs und damit ein wichtiges Argument für die Gültigkeit der Church-Turing-These, wissen um die Grenzen der Berechenbarkeit, können die Unentscheidbarkeit des Halteproblems nachweisen und durch die Reduktionsmethode auf weitere Probleme übertragen, werden durch den Nachweis der Existenz universeller Maschinen und vollständiger aufzählbarer Probleme beispielhaft an Methoden und Fragestellungen der Berechenbarkeitstheorie herangeführt, können Probleme hinsichtlich deren Zeit- und Platzkomplexität beschreiben und erhalten durch die Hierarchiesätze einen Einblick in die Auswirkungen unterschiedlicher Zeit- und Platzschranken, kennen die Grenzen der tatsächlichen Berechenbarkeit, die Klassen P und NP und das P-NP-Problem, können die NP-Vollständigkeit des Erfüllbarkeitsproblem nachweisen und durch die Reduktionsmethode auf weitere Probleme übertragen und diese damit als vermutlich nicht effizient entscheidbar charakterisieren, kennen grundlegende Begriffe der Theorie der Formalen Sprachen und können die in der Informatik betrachteten Sprachen gemäßen Stufen der Chomsky-Hierarchie als reguläre, kontextfreie, kontextsensitive und allgemeine Chomsky-Sprachen charakterisieren und die verschiedenen Stufen jeweils durch spezielle Typen von generativen Grammatiken und durch Automatenmodelle beschreiben.</p>	

<b>Inhalt</b>	Die Vorlesung gibt eine Einführung in drei zentrale Gebiete der Theoretischen Informatik: in die Berechenbarkeitstheorie, in die Komplexitätstheorie sowie in die Theorie Formaler Sprachen und die zugehörige Automatentheorie.
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse aus Mathematik und Informatik
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung
<b>Nuetzliche Literatur</b>	

## Datenbanken 1

<b>Code</b> IDB1	<b>Name</b> Datenbanken 1	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	<p>Die Studierenden sind in der Lage, eine Anforderungsanalyse und die Modellierung eines entsprechenden Datenbankschemas mit Hilfe des ER-Modells oder UML durchzuführen.</p> <p>sind in der Lage, ein Datenbankschema in einem relationalen Datenbankmanagementsystem (DBMS) zu entwickeln und zu implementieren</p> <p>sind in der Lage (komplexe) SQL Anfragen an relationale Datenbanken zu formulieren und zu evaluieren</p> <p>kennen die Techniken und Prinzipien der Anfragebearbeitung und ?optimierung</p> <p>wissen, wie Integritätsbedingungen zu identifizieren, zu formulieren und zu implementieren sind</p> <p>haben ein Verständnis von den Transaktionskonzepten und -verarbeitungsmodellen in relationalen Datenbanken</p> <p>kennen die grundlegenden Prinzipien des physischen Datenbankentwurfs und verstehen, wie diese in Anwendungen umzusetzen sind</p> <p>haben die Fähigkeit, ein weit verbreitetes DBMS (PostgreSQL oder MySQL) im Rahmen des Datenbankentwurfs und der Anfrageverarbeitung zu benutzen</p>	



<b>Inhalt</b>	<p>Architektur und Funktionalität von Datenbankmanagementsystemen (DBMS)</p> <p>Konzeptioneller Datenbankentwurf (ER-Modell und UML)</p> <p>Das relationale Datenbankmodell und relationale Anfragesprachen (Relationale Algebra, Tupel- und Domänenkalkül)</p> <p>Relationale Entwurfstheorie</p> <p>Die Anfrage- und Schemadefinitionssprache SQL</p> <p>Datenintegrität und Integritätsüberwachung, Datenbank-Trigger</p> <p>Physische Datenorganisation</p> <p>Anfragebearbeitung und ?optimierung</p> <p>Transaktionsverwaltung und Fehlerbehandlung</p> <p>Mehrbenutzersynchronisation</p> <p>Sicherheitsaspekte von Datenbanken</p> <p>Datenbankprogrammierung</p>
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
<b>Nuetzliche Literatur</b>	Alfons Kemper, André. Eickler: Datenbanksysteme - Eine Einführung, 7. Auflage, Oldenbourg Verlag, 2009.

## Einführung in Software Engineering

<b>Code</b> ISW	<b>Name</b> Einführung in Software Engineering	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 2 SWS, Blockpraktikum (2 Wochen)	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium Vorlesung und Übung 65 h Präsenzstudium Blockpraktikum 100 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen) und Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	<p>Verständnis für die Beteiligten und den Prozess der Softwareentwicklung            Kenntnis wichtiger Techniken für Anforderungsdefinition, Architekturdefinition, Entwurf, Qualitätssicherung, Wissensmanagement, Projektmanagement            Fähigkeit zur Beschreibung von Softwaresystemen auf verschiedenen Abstraktionsebenen            Fähigkeit zur Einarbeitung in komplexen objektorientierten Code</p> <p>Fähigkeit zur systematischen Erweiterung eines komplexen Systems (Anforderungen, Entwurf, Implementierung, Qualitätssicherung)            Kenntnis wichtiger Vorgehensmodelle            Fähigkeit zur Programmierung in JAVA            Umgang mit einer komplexen Entwicklungsumgebung            Umgang mit UML und CASE-Werkzeugen</p>	

<b>Inhalt</b>	<p>Die Lehrveranstaltung führt in die Entwicklung von Software im Großen ein. Sie vermittelt die Grundlagen der Modellierung und gibt eine Einführung in die wesentlichen Aktivitäten der Softwaresystementwicklung.</p> <p>Diese Aktivitäten werden in den Übungen bei der Erweiterung eines komplexen Softwaresystems durchgeführt.</p> <p>Modellierung mit der Unified Modeling Language  Überblick Softwareentwicklungsprozess, insbesondere auch Musterverwendung  Requirements Engineering: insbesondere Aufgabenbeschreibung, Datenmodellierung, Use Cases, Benutzungsschnittstellenbeschreibung  Entwurf: Analyse- und Entwurfsklassen, Architektur  Implementierung in JAVA mit einer komplexen Entwicklungsumgebung (z.B. Eclipse)  Qualitätsmanagement: Für Produkt und Prozess, Testtechniken, Inspektionstechniken, Metriken  Evolution: Wiederverwendbarkeit und Weiterentwicklung  Wissensmanagement, insbesondere Rationale  Projektmanagement  Nutzung von UML und CASE-Werkzeugen</p>
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an der Übung und dem Blockpraktikum sowie mündliche bzw. schriftliche Prüfung je nach Anzahl der TeilnehmerInnen
<b>Nuetzliche Literatur</b>	Überblick z.B. in I. Sommerville, Software Engineering, Pearson Studium oder J. Ludewig, H. Lichter, Software Engineering, dpunkt Verlag. Weitere Literatur in der Vorlesung

## Seminar

<b>Code</b> IS	<b>Name</b> Seminar	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Seminar 2 + 2 SWS (Seminar/Tutorium)	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 90 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziele</b>	<p>Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche) Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen</p> <p>Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren</p> <p>Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben</p> <p>Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen</p>	
<b>Inhalt</b>	<p>Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens</p> <p>Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur</p> <p>Fortgeschritteneres Informatikthema</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
<b>Prüfungsmodalitäten</b>	<p>Regelmäßige Teilnahme an den Seminarterminen und Mitwirkung in den Diskussionen,</p> <p>Ausarbeitung und Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion),</p> <p>schriftliche Ausarbeitung von ca. 10 Seiten</p>	
<b>Nützliche Literatur</b>		

## Fortgeschrittenenpraktikum

<b>Code</b> IFP	<b>Name</b> Fortgeschrittenenpraktikum	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b>	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Praktikum 6 SWS	<b>Arbeitsaufwand</b> 240 h; davon mind. 25 h Präsenzzeit 10 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	<p>Die Studierenden erlangen vertiefende Problemlösungskompetenz für komplexe Entwurfs- und Implementierungsaufgaben können Problemanalyse- und Beschreibungstechniken klar darstellen, differenzieren und anwenden vertiefen Programmierkenntnisse in der jeweiligen für das Projekt erforderlichen Programmiersprache sind in der Lage, das Projekt mit Hilfe einer Softwareentwicklungsumgebung durchzuführen</p> <p>Zusätzlich werden die projektypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden):</p> <p>Durchführung und Evaluation von Projekten und ihrer Phasenstruktur</p> <p>Planung und Durchführung von Projekt- und Teamarbeit.</p> <p>Zu den zu trainierenden Softskills zählen somit insbesondere Teamfähigkeit, Verfeinerung von Präsentationstechniken, etwaige Erschließung wissenschaftlicher Literatur sowie eigenverantwortliches Arbeiten.</p>	
<b>Inhalt</b>	<p>Domänenkenntnisse abhängig von den DozentInnen; allgemeine Lerninhalte sind:</p> <p>Vertiefung in die Projektarbeit</p> <p>Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>	
<b>Voraussetzungen</b>	empfohlen sind: Anfängerpraktikum (IAP), Einführung in Software Engineering (ISW)	
<b>Pruefungsmodalitaeten</b>	Bewertung der dokumentierten Software, des Projektberichts (5-10 Seiten) und des Vortrags (ca. 30 Minuten zzgl. Diskussion)	
<b>Nuetzliche Literatur</b>		

## 2.2 Pflichtmodule Mathematik

Die Vermittlung der mathematischen Grundlagen erfolgt durch das Modul *Mathematik für Informatiker 1*, welches nachfolgend beschrieben ist. Wenn im anderen Hauptfach bereits ein Mathematikmodul im Grundlagenbereich absolviert wurde, so kann dieses Modul hier entfallen, und stattdessen ein Wahlpflichtmodul im Umfang von 8 LP aus dem Wahlpflichtbereich des Bachelor-Studienganges Angewandte Informatik gewählt werden. Zur Wahl stehen dabei Module aus dem Kapitel 3.2 Wahlpflichtmodule Informatik und aus dem Kapitel 3.4 Wahlpflichtmodule Technische Informatik des Modulhandbuchs des Bachelor-Studienganges Angewandte Informatik 100% sowie Module aus dem Kapitel 2.3 Module aus der Informatik und aus dem Kapitel 2.6 Module aus dem M.Sc. Technische Informatik des Modulhandbuchs des Master-Studienganges Angewandte Informatik.

## Mathematik für Informatiker 1

<b>Code</b> IMI1	<b>Name</b> Mathematik für Informatiker 1	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 20 h Prüfungsvorbereitung 130 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik
<b>Lernziele</b>	Hinführung zu mathematischen Denkweisen (Abstrahieren, Strukturieren), theoretisch fundiertes Verständnis und praktische Beherrschung einfacher Rechenverfahren aus der Linearen Algebra insbesondere mit Blick auf Anwendungen in der Informatik	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- EINFÜHRUNG: Symbolsprache der Mathematik, logische Verknüpfungen (Aussagenlogik), Beweisarten, Mengen, Relationen, Abbildungen, grundlegende algebraische Strukturen</li> <li>- VEKTORRÄUME: Unterräume, Basis, Dimension, Koordinaten, Anwendungen in Geometrie und Computergrafik.</li> <li>- LINEARE ABBILDUNGEN: Kern (Nullraum), Bild(raum), Matrizen, Rang, Determinanten, charakteristisches Polynom, Eigenwerte und Eigenräume, Diagonalisierung von Matrizen, lineare Gleichungssysteme, elementare Lösungsverfahren und Eigenschaften, Anwendungen in der Datenanalyse.</li> <li>- INNENPRODUKTRÄUME: Bilinearformen, Orthogonalität, Orthonormalbasen, selbstadjungierte, isometrische (und normale) Operatoren, Spektralsätze, Ausblick zum wissenschaftlichen Rechnen.</li> </ul>	
<b>Voraussetzungen</b>	empfohlen ist: Schulwissen in Mathematik	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung. Es wird am Ende der Vorlesungszeit eine Klausur angeboten. Wird diese nicht bestanden so kann die Prüfungsleistung in einer zweiten Klausur vor Beginn der nächsten Vorlesungszeit erbracht werden.	
<b>Nützliche Literatur</b>		

## 2.3 Pflichtmodule Fachdidaktik

Nachfolgend werden die Module *Fachdidaktik 1 und 2* beschrieben. Das Modul *Fachdidaktik 1* besteht aus zwei Teilen, welche unabhängig voneinander absolviert werden können. Die beiden Teile sind *Didaktik der ITG* und *Didaktik der Informatik*, welche jeweils ein Seminar sind.



## Fachdidaktik 1

<b>Code</b> IFD1	<b>Name</b> Fachdidaktik 1	
<b>Leistungspunkte</b> insg. 4 LP (2 x 2 LP)	<b>Dauer</b> 2 unabhängige Teile zu je einem Semester	<b>Turnus</b> pro Semester mind. ein Teil
<b>Lehrform</b> 2 unabhängige Seminare zu je 2 SWS	<b>Arbeitsaufwand</b> 2 Seminare zu jeweils 60h; davon  30 h Präsenzstudium, 15 h Vor- und Nachbereitung, 15 h Verfassen der Hausarbeit	<b>Verwendbarkeit</b> Lehramt Informatik
<b>Lernziele</b>	<p>Die Studierenden...</p> <p>... können Bildungsziele der Informatik in den Allgemeinbildungsauftrag der Schule einordnen.</p> <p>... haben Einblick in fachdidaktische Konzepte zur Vermittlung informationstechnischer Grundbildung und informatischer Kompetenzen und kennen Methoden zum Entwurf von Unterrichtseinheiten.</p> <p>... können Aufgabenstellungen altersgerecht aufbereiten, in die Erfahrungswelt der Schüler/-innen übertragen und einen teamorientierten Unterricht gestalten.</p> <p>... können informatikspezifische Curricula vergleichen und zugehörige Unterrichtspläne in attraktive konsekutive Unterrichtseinheiten umsetzen.</p> <p>... sind mit den einschlägigen Ergebnissen der Lehr-Lernforschung vertraut und in der Lage, konzeptionelle Entwürfe vor dem Hintergrund aktueller Erkenntnisse zu reflektieren.</p> <p>... kennen im ITG-Unterricht und Informatikunterricht einsetzbare Werkzeuge und Systeme.</p>	
<b>Inhalt</b>	<p>Grundlegende Inhalte der Fachdidaktik ITG und Fachdidaktik Informatik:</p> <ul style="list-style-type: none"> <li>- Bildungsziele der Informatik; Kernaspekte informationstechnischer Grundbildung; Begründung für den Informatikunterricht; Charakterisierung des Fachs und fundamentale Ideen; Auswahlkriterien für Unterrichtsinhalte</li> <li>- Lehr-Lernprozesse inklusive Lernvoraussetzungen und Lernschwierigkeiten</li> <li>- Methoden des ITG-Unterrichts und des Informatikunterrichts, insbesondere Auswahl und Einsatz von Werkzeugen, Projektarbeiten und Vorgehensweisen bei der Erfolgskontrolle</li> </ul>	

<b>Voraussetzungen</b>	empfohlen sind: grundlegende Inhalte und Methoden der Informatik
<b>Pruefungsmodalitaeten</b>	In beiden Teilen wird eine Hausarbeit verfasst, die jeweils zu 50% in die Modulnote eingeht.
<b>Nuetzliche Literatur</b>	Hartmann, W., Näf, M. & Reichert, R. (2006). Informatikunterricht planen und durchführen. Springer. Hubwieser, P. (2007). Didaktik der Informatik: Grundlagen, Konzepte, Beispiele (3., überarb. u. erw. Aufl.). Springer. Schubert, S. & Schwill, A. (2011). Didaktik der Informatik (2. Aufl.). Spektrum Akademischer Verlag.

## Fachdidaktik 2

<b>Code</b> IFD2	<b>Name</b> Fachdidaktik 2	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung/ Seminar 2 SWS	<b>Arbeitsaufwand</b> 180h; davon 20 h Präsenzstudium, 10 h Hospitation Schule, 50 h Vorbereitung, 100 h Planung und Dokumentation Unterrichtseinheit	<b>Verwendbarkeit</b> Lehramt Informatik
<b>Lernziele</b>	<p>Die Studierenden...</p> <ul style="list-style-type: none"> <li>... können Bildungsziele der Informatik in den Allgemeinbildungsauftrag der Schule einordnen.</li> <li>... haben Einblick in fachdidaktische Konzepte zur Vermittlung informatischer Kompetenzen und der didaktischen Reduktion von Fachinhalten</li> <li>... kennen Methoden zum Entwurf von Unterrichtseinheiten, wenden diese praktisch an und reflektieren sie.</li> <li>... können Aufgabenstellungen alters- und Zielgruppengerecht aufbereiten, in die Erfahrungswelt der Schüler/-innen übertragen.</li> <li>... können informatikspezifische Curricula vergleichen, Inhalte geeignet auswählen und zugehörige Unterrichtspläne in attraktive konsekutive Unterrichtseinheiten mit Berücksichtigung der Lerngruppe umsetzen.</li> <li>... sind in der Lage, konzeptionelle Entwürfe vor dem Hintergrund aktueller Erkenntnisse sowie die eigene Planung und Durchführung von Unterricht zu reflektieren.</li> <li>... kennen im Informatikunterricht einsetzbare Werkzeuge und Systeme und können eine geeignete und begründete Auswahl treffen.</li> </ul>	

<b>Inhalt</b>	<p>Die Veranstaltung beschäftigt sich insbesondere mit didaktisch-methodischen Aspekten der Gestaltung von Unterrichtseinheiten im Informatikunterricht und führt in die praktische Planung ein. Es findet eine Hospitations- und Praxisphase in Kooperation mit lokalen Schulen statt, in der die eigene Unterrichtsplanung erprobt und reflektiert wird.</p> <p>Zentrale Inhalts- und Prozesskonzepte der Informatik / fundamentale Ideen , allgemeinbildender Informatikunterricht. Besonderheiten des Informatikunterrichts organisatorischer und inhaltlicher Art.</p> <p>Methoden des Informatikunterrichts, insbesondere Auswahl und Einsatz von geeigneten Inhalten und Werkzeugen, didaktische Reduktion, spezifische Arbeitsformen im Informatikunterricht, Projektarbeiten, Binnendifferenzierung, Lernvoraussetzungen, modularer Aufbau von Unterrichtseinheiten und Vorgehensweisen bei der Erfolgskontrolle.</p>
<b>Voraussetzungen</b>	empfohlen sind: Fachdidaktik 1, Teil 1 und 2
<b>Pruefungs-modalitaeten</b>	Vortrag und schriftliche Ausarbeitung
<b>Nuetzliche Literatur</b>	<p>Didaktik der Informatik, Peter Hubwieser, Springer, 2007</p> <p>W. Hartmann, M. Näf, R. Reichert: Informatikunterricht planen und durchführen, Springer 2006</p> <p>Schubert, S. &amp; Schwill, A. (2011). Didaktik der Informatik (2. Aufl.). Spektrum Akademischer Verlag.</p>

# 3 Wahlpflichtbereich

Zur Zeit werden keine Wahlpflichtmodule ausschließlich für den Lehramt-Studiengang Informatik angeboten. Zur Wahl stehen Module aus dem Kapitel 3.2 Wahlpflichtmodule Informatik und aus dem Kapitel 3.4 Wahlpflichtmodule Technische Informatik des Modulhandbuchs des Bachelor-Studienganges Angewandte Informatik 100% sowie Module aus dem Kapitel 2.3 Module aus der Informatik und aus dem Kapitel 2.6 Module aus dem M.Sc. Technische Informatik des Modulhandbuchs des Master-Studienganges Angewandte Informatik.