

2 Pflichtbereich

Im Folgenden sind die Pflichtmodule des Bachelor-Studiengangs Angewandte Informatik mit einem Fachanteil von 50% beschrieben, unterteilt in die Bereiche Informatik und Mathematik.

2.1 Pflichtmodule Informatik

Nachfolgend sind die Pflichtmodule der Informatik beschrieben. Die Reihenfolge der Module orientiert sich dabei an der Abfolge im Studienverlaufsplan auf Seite 6.

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.

Das Modul Bachelor-Arbeit ist nur dann ein Pflichtmodul, wenn Angewandte Informatik das 1. Hauptfach ist, also die Bachelor-Arbeit hier angefertigt wird. Ist Angewandte Informatik das 2. Hauptfach, so entfällt dieses Modul.

Einführung in die Praktische Informatik

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

Pruefungs- modalitaeten	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.
Nuetzliche Literatur	Wird jährlich aktualisiert

Programmierkurs

Code IPK	Name Programmierkurs	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus jedes Semester
Lehrform Praktikum 2 SWS	Arbeitsaufwand 90 h; davon 30 h Präsenzstudium 30 h praktische Übung am Rechner 30 h Hausaufgaben	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	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	
Inhalt	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.	
Voraussetzungen	keine	
Pruefungs-modalitaeten	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.	
Nuetzliche Literatur		

Programmierkurs mit Python

Code IPK-Python	Name Programmierkurs mit Python	
Leistungspunkte 3 LP	Dauer ein Semester	Turnus jedes Semester
Lehrform Praktikum 2 SWS	Arbeitsaufwand 90 h; davon 30 h Präsenzstudium 30 h praktische Übung am Rechner 30 h Hausaufgaben	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	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.	
Inhalt	<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>	
Voraussetzungen	keine	

Pruefungs- modalitaeten	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.
Nuetzliche Literatur	

Algorithmen und Datenstrukturen

Code IAD	Name Algorithmen und Datenstrukturen	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 4 SWS, Übung 2 SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	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	
Inhalt	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)	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)	

Pruefungs- modalitaeten	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
Nuetzliche Literatur	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

Einführung in die Theoretische Informatik

Code ITH	Name Einführung in die Theoretische Informatik	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 4 SWS, Gruppen-Übung 2 SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	<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>	

Inhalt	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.
Voraussetzungen	empfohlen sind: Grundkenntnisse aus Mathematik und Informatik
Pruefungs- modalitaeten	Erfolgreiche Teilnahme an den Gruppenübungen und erfolgreiche Teilnahme an einer schriftlichen Prüfung
Nuetzliche Literatur	

Einführung in die Technische Informatik

Code ITE	Name Einführung in die Technische Informatik	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Wintersemester
Lehrform Vorlesung 4 SWS, Übung 2 SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	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	
Inhalt	Schaltalgebra Digitale Schaltungen Sequentielle Logik Technologische Grundlagen Programmierbare Logikbausteine Zahlendarstellung und Codierung Rechnerarithmetik Ein einfacher Prozessor Pipelineverarbeitung von Befehlen Vorhersage von Sprüngen Peripherie	
Voraussetzungen	keine	
Pruefungs- modalitaeten	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>Nuetzliche Literatur</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 & Architecture.Addison-Wesley, 2001.</p>
--	--

Betriebssysteme und Netzwerke

Code IBN	Name Betriebssysteme und Netzwerke	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 4 SWS, Gruppen-Übung 2 SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	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.	
Inhalt	<p>Themen der Betriebssystemtechnik sind:</p> <ul style="list-style-type: none"> * Prozesse und ihre Verwaltung * Verwaltung des Speichers im Rechner * Prozesssynchronisation * Nebenläufigkeit und Verklemmungen * Scheduling * Eingabe/Ausgabe und Dateiverwaltung <p>Themen der Netzwerktechnik sind:</p> <ul style="list-style-type: none"> * Schichtenmodell der Rechnerkommunikation * Direktverbindungsnetze * Paketvermittlung * Internetworking * Ende-zu-Ende-Protokolle * Überlastkontrolle * Anwendungen 	
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI)	
Prüfungsmodalitäten	Erfolgreiche Teilnahme an den Übungen und erfolgreiche Teilnahme an einer schriftlichen oder mündlichen Abschlussprüfung je nach Anzahl der TeilnehmerInnen	

Nuetzliche Literatur	<ul style="list-style-type: none">* 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 Software Engineering

Code ISW	Name Einführung in Software Engineering	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Wintersemester
Lehrform Vorlesung 3 SWS, Übung 2 SWS, Blockpraktikum (2 Wochen)	Arbeitsaufwand 240 h; davon 75 h Präsenzstudium Vorlesung und Übung 65 h Präsenzstudium Blockpraktikum 100 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen) und Prüfungsvorbereitung	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	<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>	

Inhalt	<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>
Voraussetzungen	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)
Pruefungs-modalitaeten	Erfolgreiche Teilnahme an der Übung und dem Blockpraktikum sowie mündliche bzw. schriftliche Prüfung je nach Anzahl der TeilnehmerInnen
Nuetzliche Literatur	<p>Überblick z.B. in I. Sommerville, Software Engineering, Pearson Studium oder J. Ludewig, H. Lichter, Software Engineering, dpunkt Verlag.</p> <p>Weitere Literatur in der Vorlesung</p>

Datenbanken 1

Code IDB1	Name Datenbanken 1	
Leistungspunkte 8 LP	Dauer ein Semester	Turnus jedes Sommersemester
Lehrform Vorlesung 4 SWS, Übung 2 SWS	Arbeitsaufwand 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik
Lernziele	<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>	

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

Seminar

Code IS	Name Seminar	
Leistungspunkte 4 LP	Dauer ein Semester	Turnus jedes Semester
Lehrform Seminar 2 + 2 SWS (Seminar/Tutorium)	Arbeitsaufwand 120 h; davon 30 h Präsenzstudium 90 h Vorbereitung Vortrag	Verwendbarkeit B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
Lernziele	<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>	
Inhalt	<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>	
Voraussetzungen	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
Prüfungsmodalitäten	<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>	
Nützliche Literatur		

Bachelor-Arbeit

Code IBa_50	Name Bachelor-Arbeit	
Leistungspunkte 12 LP	Dauer	Turnus jedes Semester
Lehrform Betreutes Selbststudium 1 SWS, Kolloquium 1 SWS	Arbeitsaufwand 360 h; davon 320 h Bearbeitung eines individuellen Themas (Forschungs- und Entwicklungsarbeiten) und schriftliche Ausarbeitung 40 h Ausarbeitung Vortrag und Präsentation und Mitwirkung Kolloquium	Verwendbarkeit B.Sc. Angewandte Informatik mit einem Fachteil von 50%
Lernziele	Einsatz der erlernten Fachkenntnisse und Methoden zum selbstständigen Lösen einer überschaubaren Problemstellung aus der Informatik und ihren Anwendungen Fähigkeit, eine wissenschaftlichen Arbeit zu erstellen Fähigkeit, eigene Arbeiten in einem wissenschaftlichen Vortrag darzustellen	
Inhalt	selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung	
Voraussetzungen	nach Prüfungsordnung insgesamt mindestens 120 LP, wovon auf den Bereich Informatik mindestens 60 LP entfallen; weiterhin sind empfohlen die Module Seminar (IS) und Anfängerpraktikum (IAP)	
Pruefungs- modalitaeten	regelmäßige Treffen mit der/dem BetreuerIn, schriftliche Ausarbeitung und Präsentation	
Nuetzliche Literatur		