

Modulkatalog

Bachelor-Studiengang Angewandte Informatik Prüfungs- und Studienordnung 2016

***Fachbereich Information und Kommunikation
Hochschule Flensburg***

Hinweis:

Da sich die Modulbeschreibungen in erster Linie an Studierende bzw. Studieninteressierte richten, sind die Kompetenzen in den Lernzielen in persönlicher Ansprache mit "Sie" formuliert, etwa: "Darüber hinaus sind Sie in der Lage, ..."

Inhaltsverzeichnis

3D-Computergraphik	4
Algorithmen	5
Anwendungsprogrammierung	6
Bachelor-Arbeit	7
Berufspraktikum	8
Betriebswirtschaftslehre	9
Computernetze	10
Computerarchitektur und Betriebssystem	11
Datenbanken	12
Digitale Signalverarbeitung	13
Fachenglisch	14
Funktionale Programmiersprachen	15
Informatik-Seminar	16
Interface- und Interaktionsdesign	17
IT-Sicherheit	18
Kommunikation und Präsentation	19
Konzepte moderner Programmiersprachen	20
Kryptografie	21
Mathematik 1	22
Mathematik 2	23
Medizinische Visualisierung	24
Mobile Computing	25
Netzwerk-Administration	26
Objektorientierte Programmierung	27
Plugin-Entwicklung mit Visual Studio	28
Projekt	29
Projektmanagement	30
Recht	31
Software Engineering 1	32
Software Engineering 2	33
Strukturierte Programmierung	34
Systeme der Mobilkommunikation	35
Systemprogrammierung	36
Theoretische Informatik	37
Usability Engineering	38
Verteilte Webanwendungen und Webservices	39
Web-Programmierung	40
Web-Systeme	41
Webdesign	42
Webtechnologien	43

3D-Computergraphik

Modulbezeichnung:	3D-Computergraphik
Modulverantwortlicher:	Prof. Dr. K. Hartmann
Lehrender:	Prof. Dr. K. Hartmann
Semester:	5./6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Workshop, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	
Inhalt:	<p>Die Veranstaltung gibt eine Einführung in die Programmierung mit OpenGL.</p> <p>OpenGL ist eine plattformübergreifende Grafik-Bibliothek. Besonders interessant ist die Variante OpenGL ES, die auf aktuellen mobilen Plattformen eingesetzt wird, aber auch Grundlage für WebGL ist. OpenGL ES unterstützt iDevices ab der 3. Generation 2.0, womit programmierbare Shader möglich sind.</p> <p>Für eigene Experimente in vorlesungsbegleitenden Projekten können sowohl mobile Plattformen als auch Workstations genutzt werden.</p> <p>Die Veranstaltung setzt keine Vorkenntnisse voraus und zielt auf ein praktisches Verständnis der zu Grunde liegenden mathematischen Formalismen. Das vorlesungsbegleitende Projekt ist daher im Umfang relativ klein und wird in kleinen Teams durchgeführt.</p>
Prüfungsform:	PL: Hausarbeit
Literatur:	

Algorithmen

Modulbezeichnung:	Algorithmen
Modulverantwortlicher:	Prof. Dr. J. Christiansen
Lehrender:	Prof. Dr. J. Christiansen
Semester:	3. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Informatik-Grundlagen)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie können Methoden zum Entwurf von Algorithmen anwenden, die Korrektheit von Algorithmen beweisen, Algorithmen hinsichtlich ihrer Laufzeit analysieren und Algorithmen in Programme umsetzen. Ferner kennen Sie die wichtigsten fundamentalen Algorithmen und Datenstrukturen.
Inhalt:	<p>Vorlesung</p> <p>Fundamentale Algorithmen</p> <ul style="list-style-type: none"> • Sortieren: Insertionsort, Quicksort, Heapsort • Textsuche: Algorithmus von Knuth-Morris-Pratt, Algorithmus von Boyer-Moore • Graphen: Zusammenhangskomponenten, minimaler Spannbaum, kürzeste Wege <p>Analyse</p> <ul style="list-style-type: none"> • O-Notation <p>Entwurfsmethoden</p> <ul style="list-style-type: none"> • Divide and Conquer • Greedy • Dynamic Programming <p>Algorithmisch schwierige Probleme</p> <ul style="list-style-type: none"> • Problem des Handlungsreisenden • Klassen P und NP, NP-Vollständigkeit • Approximationsverfahren <p>Übungen / Labor</p> <p>In den begleitenden Übungen werden ausgewählte Algorithmen am Computer programmiert und empirisch hinsichtlich ihrer Laufzeiten verglichen.</p>
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	T.H. CORMEN, C.E. LEISERSON, R.L. RIVEST, C. STEIN: Introduction to Algorithms. 2. Auflage, The MIT Press (2001) H.W. LANG: Algorithmen in Java. 3. Auflage, Oldenbourg (2012) R. SEDGEWICK, K. WAYNE: Algorithms. 4. Auflage, Addison Wesley (2011)

Anwendungsprogrammierung

Modulbezeichnung:	Anwendungsprogrammierung
Modulverantwortlicher:	O. Preikszas, B.Sc.
Lehrender:	O. Preikszas, B.Sc.
Semester:	3. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Softwaretechnik)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	<p>Sie können Anwendungen mit grafischen Benutzerschnittstellen konzipieren und erstellen. Dabei können Sie die Unterschiede verschiedener Frameworks und Sprachen einschätzen und entsprechend nutzen.</p> <p>Sie sind in der Lage, komplexere Datenstrukturen programmtechnisch umzusetzen und sinnvoll einzusetzen.</p>
Inhalt:	<ul style="list-style-type: none"> • User Interfaces: Paradigmen und Metaphern • Konzepte verschiedener User-Interface-Frameworks: Swing, .NET Forms, Qt, TKinter • Deklarative <i>cross-platform</i>-Systeme, z.B. <u>QML</u>, <u>XUL</u>, <u>XAML</u> und <u>WPF</u> • Entwicklung komplexerer Datenstrukturen (Bäume, Hashtabellen), Verwendung entsprechender Collection- bzw. Container-Klassenbibliotheken <p>Übungen</p> <p>In den Laborübungen werden Programmieraufgaben zu o.g. Themen gestellt. Die Lösungen werden analysiert und diskutiert.</p>
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	

Bachelor-Arbeit

Modulbezeichnung:	Bachelor-Arbeit
Modulverantwortlicher:	N.N.
Lehrender:	N.N.
Semester:	7. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach
Lehrform:	
Arbeitsaufwand:	Präsenzstudium: 10 h, Eigenstudium: 350 h, Gesamtaufwand: 360 h
Leistungspunkte:	12 cp
Voraussetzungen:	siehe Prüfungs- und Studienordnung
Lernziele / Kompetenzen:	In der Bachelor-Arbeit sollen Sie zeigen, dass Sie in der Lage sind, ein Problem Ihrer Fachrichtung selbstständig auf wissenschaftlicher Grundlage methodisch zu bearbeiten.
Inhalt:	<p>Die Bachelor-Arbeit ist eine das Bachelor-Studium abschließende Prüfungsarbeit. Das Thema der Arbeit können Sie selbst vorschlagen – meist ergibt es sich im vorausgehenden Berufspraktikum.</p> <p>Sie werden während der Anfertigung der Bachelorarbeit von einem oder einer Lehrenden der Hochschule betreut.</p> <p>Die Bearbeitungszeit für die Bachelor-Arbeit beträgt zwei Monate.</p> <p>Zur Bachelor-Prüfung gehört ein 45-minütiges Kolloquium, in dem Sie die Ergebnisse Ihrer Arbeit erläutern.</p>
Prüfungsform:	PL: Schriftliche Ausarbeitung, Kolloquium
Literatur:	

Berufspraktikum

Modulbezeichnung:	Berufspraktikum
Modulverantwortlicher:	N.N.
Lehrender:	N.N.
Semester:	7. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach
Lehrform:	Praktikum
Arbeitsaufwand:	Präsenzstudium: 480 h, Eigenstudium: 60 h, Gesamtaufwand: 540 h
Leistungspunkte:	18 cp
Voraussetzungen:	siehe Prüfungs- und Studienordnung
Lernziele / Kompetenzen:	<p>Im Berufspraktikum werden Sie an die ingenieurmäßige Tätigkeit der Informatikerin oder des Informatikers herangeführt. Dies geschieht durch praktische und projektbezogene Mitarbeit in unterschiedlichen Aufgaben- und Verantwortungsbereichen des Betriebs. Dadurch wird eine enge Verbindung zwischen Studium und Berufspraxis hergestellt. Sie erhalten Einblick in betriebliche Abläufe vom Auftragseingang bis zur Auslieferung. Im Vordergrund steht nicht der Erwerb von Detailwissen, sondern das Erfassen von betrieblichen Gesamtzusammenhängen.</p>
Inhalt:	<p>Das dreimonatige Berufspraktikum absolvieren Sie in einem Betrieb Ihrer Wahl. Sie werden dabei von einem Ansprechpartner im Betrieb fachlich betreut.</p> <p>Vor Aufnahme des Praktikums nehmen Sie an einem Einführungsseminar an der Hochschule teil.</p> <p>Nach Beendigung des Praktikums berichten Sie in einem Abschlusssseminar an der Hochschule über Ihr Praktikum und über die Ergebnisse eines interessanten Projekts, das Sie während Ihres Praktikums im Betrieb durchgeführt haben.</p>
Prüfungsform:	SL: Abschlusssseminar, Praktikumsbericht
Literatur:	

Betriebswirtschaftslehre

Modulbezeichnung:	Betriebswirtschaftslehre
Modulverantwortlicher:	Prof. Dr. K. Pfaffenberger
Lehrender:	Prof. Dr. K. Pfaffenberger
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Übergreifende Qualifikationen)
Lehrform / SWS:	Vorlesung, 2 SWS
Arbeitsaufwand:	Präsenzstudium: 30 h, Eigenstudium: 45 h, Gesamtaufwand: 75 h
Leistungspunkte:	2.5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie sind in der Lage, grundlegende ökonomische Probleme zu erkennen und zu analysieren. Sie beherrschen wichtige Instrumente der Erfolgskontrolle.
Inhalt:	<p>Einführung in die Wirtschaftswissenschaften</p> <ul style="list-style-type: none"> • ökonomische Grundbegriffe • Das Unternehmen im volkswirtschaftlichen Zusammenhang <p>Unternehmen und Märkte</p> <ul style="list-style-type: none"> • Betriebswirtschaftliche Kategorien (Kosten, Gewinn, Rentabilität, Produktivität) • Angebots- und Nachfrageverhalten • Preismechanismus und Gleichgewicht auf den Märkten <p>Ziele unternehmerischer Aktivitäten und Erfolgskontrolle</p> <ul style="list-style-type: none"> • ROI-Baum • Kurzfristige Erfolgsrechnung mittels Deckungsbeiträgen • Break-Even-Analyse • Investitionsrechenverfahren • Strategische Konzepte der Erfolgsmessung (z.B. Portfolio-Analyse)
Prüfungsform:	SL: Klausur oder sonstige Prüfungsleistung
Literatur:	<p>H. SCHECK, B. SCHECK: Wirtschaftliches Grundwissen für Naturwissenschaftler und Ingenieure. 2. Auflage, Wiley-VCH (2007)</p> <p>G. WÖHE, U. DÖRING: Einführung in die Allgemeine Betriebswirtschaftslehre. 22. Auflage, Vahlen (2005)</p>

Computernetze

Modulbezeichnung:	Computernetze
Modulverantwortlicher:	Prof. Dr. S. Gajek
Lehrender:	Prof. Dr. S. Gajek
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Computernetze und IT-Sicherheit)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie kennen die grundlegenden Konzepte des Internet. Sie sind mit den Protokollen nach dem ISO/OSI-Modell vertraut und kennen entsprechende Referenzimplementierungen. Sie wissen, wie kabelgebundene und drahtlose Kommunikation funktioniert, welche Koppellelemente für die Verbindungen von Netzen existieren und wie Nachrichtenpakete navigiert werden. Sie kennen auch die Sicherheitsschwachstellen der Internetprotokolle und zugehörige Schutzmechanismen.
Inhalt:	<p>Vorlesung</p> <p>Es werden die Grundlagen der Netzwerkkommunikation vermittelt. Am Beispiel des Internet lernen Sie die grundlegenden Mechanismen der Nachrichtenvermittlung, -flusskontrolle, und -sicherheit kennen. Anhand von Referenzimplementierungen verschiedener Protokolle lernen Sie die Funktionen auf</p> <ul style="list-style-type: none">• Applikationsschicht (z.B. <u>H</u>TTP, <u>F</u>TP, <u>P</u>OP/<u>I</u>MAP/<u>S</u>MTP, <u>D</u>NS)• Transportschicht (z.B. <u>T</u>CP, <u>U</u>DP)• Netzwerkschicht (z.B. <u>I</u>P, <u>B</u>GP, <u>R</u>IP)• Sicherungs- und Bitübertragungsschicht (z.B. Ethernet, IEEE 802.11) <p>Labor</p> <p>Sie analysieren die Protokolle mit gängigen Werkzeugen der Netzwerkanalyse. Der Fokus der Analyse liegt im Bereich der Netzwerksicherheit.</p>
Prüfungsform:	PL: Klausur oder sonstige Prüfungsleistung
Literatur:	A.S. TANENBAUM: Computernetzwerke. 5. Auflage, Pearson Studium (2012)

Computerarchitektur und Betriebssystem

Modulbezeichnung:	Computerarchitektur und Betriebssystem
Modulverantwortlicher:	B. Schulz, B.Sc.
Lehrender:	B. Schulz, B.Sc.
Semester:	1. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Computernetze und IT-Sicherheit)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie verstehen die Abläufe im Computer während der Programmausführung. Sie sind in der Lage, Computerkomponenten hinsichtlich ihrer Leistungsfähigkeit für bestimmte Einsatzgebiete und Aufgabenstellungen zu bewerten. Sie haben erste praktische Erfahrungen in der Verwaltung linux-basierter Betriebssysteme gesammelt.
Inhalt:	<p>Darstellung von Informationen und arithmetischen Operationen</p> <ul style="list-style-type: none"> • Darstellung ganzer Zahlen: Vorzeichen-Betrag, Exzess, Komplemente • Darstellung reeller Zahlen: Gleitkommazahlen im IEEE 754 Standard • Arithmetische Operationen <p>Rechnerarchitekturen</p> <ul style="list-style-type: none"> • Harvard- und von-Neumann-Architektur • Speicher-, Rechen- und Steuerwerk • Speicherorganisation • Befehls-Pipelining <p>Betriebssystem</p> <ul style="list-style-type: none"> • Variablen und Funktionen, Stack, Heap • Rechte-Management und Zugriffsschutz: Betriebssystem- und Nutzerprozesse • Thread-Management • Shell (Bash, Powershell) • Systemverwaltung: Konfiguration, Automatisierung und Prozesskontrolle
Prüfungsform:	PL: Klausur
Literatur:	A.S. TANENBAUM, T. AUSTIN: Rechnerarchitektur: Von der digitalen Logik zum Parallelrechner. 6. Auflage, Pearson Studium (2014) A.S. TANENBAUM, H. Bos: Moderne Betriebssysteme. 4. Auflage, Pearson Studium (2016)

Datenbanken

Modulbezeichnung:	Datenbanken
Modulverantwortliche:	Prof. Dr. M. Zachow
Lehrende:	Prof. Dr. M. Zachow
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Informatik-Grundlagen)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie kennen die grundlegenden Konzepte von relationalen Datenbanken und alternativen Ansätzen. Sie können eine relationale Datenbank entwerfen, den Entwurf beurteilen und mithilfe der Sprache SQL realisieren. Sie können Datenbankabfragen in SQL formulieren. Sie haben praktische Erfahrungen beim Erstellen von webbasierten Datenbank Anwendungen gesammelt.
Inhalt:	<p>Die Idee einer Datenbank ist, alle Daten nach einem einheitlichen Konzept und völlig unabhängig von den Anwendungsprogrammen zu verwalten. Das Datenbanksystem stellt für die Anwendungsprogramme einheitliche Zugriffsmethoden auf die Daten zur Verfügung. Die sehr verbreiteten relationalen Datenbanken stellen alle Daten in Form von Tabellen dar. Die Schnittstelle zu den Anwendungsprogrammen ist die sehr einfache Sprache SQL. Andere Datenbankkonzepte werden häufig mit dem Begriff NoSQL zusammengefasst.</p> <p>Vorlesung</p> <ul style="list-style-type: none">• Architektur eines Datenbanksystems• relationale Datenbank• Datenbankoperationen in SQL• Datenbank-Entwurf, Schlüssel, Normalform• objekt-relationale Abbildung (ORM)• Transaktionen• NoSQL-Datenbanken <p>Übungen / Labor</p> <p>In den begleitenden Übungen werden die Themen aus der Vorlesung angewendet: Modellierung von Datenbanken, Abfragen, Prüfung des Entwurfs. Außerdem wird eine webbasierte Datenbank Anwendung mit Java erstellt.</p>
Prüfungsform:	PL: Klausur
Literatur:	A. KEMPER, A. EICKLER: Datenbanksysteme. 8. Auflage, Oldenbourg (2011)

Digitale Signalverarbeitung

Modulbezeichnung:	Digitale Signalverarbeitung
Modulverantwortlicher:	Prof. Dr. T. Aschmoneit
Lehrender:	Prof. Dr. T. Aschmoneit
Semester:	3. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Usability und Mobile)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie kennen die Grundlagen der digitalen Signalverarbeitung. Sie kennen die wichtigsten Algorithmen digitalen Signalverarbeitung und können diese praktisch anwenden.
Inhalt:	<p>Vorlesung</p> <p>Einführung und Überblick</p> <ul style="list-style-type: none"> • Einsatzbereiche der digitalen Signalverarbeitung • Digitale Signalprozessoren • Matlab <p>Systemtheoretische Grundlagen</p> <p>FIR-Filter</p> <ul style="list-style-type: none"> • Linearphasige FIR-Filter <p>Diskrete Fourier Transformation</p> <ul style="list-style-type: none"> • Fast Fourier Transformation (DIT, Radix-2-Algorithmus) • Faltung mittels DFT • Spektralanalyse, DFT-Faltung unendlich langer Signale • Fensterung <p>Diskrete Cosinus Transformation (JPEG-Kompression)</p> <p>Korrelation</p> <p>IIR-Filter</p> <ul style="list-style-type: none"> • Filterstrukturen (Direktstrukturen, Kaskadenstrukturen) • Entwurf von IIR-Filtern • Impulsinvariante Transformation • Bilineare Transformation • Transformation von Butterworth- und Tschebycheff-Filtern <p>Labor</p> <ul style="list-style-type: none"> • Anwendung Laborsystem • Signalsynthese • FIR-Filter • Diskrete Fouriertransformation • Korrelation • Wechselnde Projekte in Gruppenarbeit, z.B. Simulation digitale Übertragung im Basisband
Prüfungsform:	PL: Klausur oder sonstige Prüfungsleistung
Literatur:	H. GÖTZ: Einführung in die digitale Signalverarbeitung. 3. Auflage, Teubner (1998) J.R. JOHNSON: Digitale Signalverarbeitung. Hanser (1991) MATHWORKS INC.: Matlab Dokumentation: Signalprocessing Toolbox. H.W. SCHÜBLER: Digitale Signalverarbeitung. 4. Auflage, Springer (1994)

Fachenglisch

Modulbezeichnung:	Fachenglisch
Modulverantwortliche:	I. Steindorf, B.A.
Lehrende:	I. Steindorf, B.A.
Semester:	1. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Übergreifende Qualifikationen)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie können englischsprachige naturwissenschaftliche Texte verstehen und naturwissenschaftliche Zusammenhänge auf Englisch formulieren. Hierfür verfügen Sie über einen entsprechenden Wortschatz und Kenntnisse der Grammatik. Sie sind in der Lage, das Gelernte in fachlichen Diskussionen, Telefongesprächen, Bewerbungen usw. anzuwenden.
Inhalt:	<ul style="list-style-type: none">• Grammatikstrukturen, z.B. Passiv, Wortstellung, Satzbau• Physikalische Begriffe: circuit, resistance, resistor, switch, speed, velocity, ...• Mathematische Begriffe: to subtract, to divide, to cancel, fraction, ...• Ausgewählte Themen aus den Bereichen: Ladungen und Felder, Schwingungen und Wellen, Quantenphysik
Prüfungsform:	SL: Sonstige Prüfungsleistung
Literatur:	

Funktionale Programmiersprachen

Modulbezeichnung:	Funktionale Programmiersprachen
Modulverantwortliche:	Dr. P. Sadeghi
Lehrende:	Dr. P. Sadeghi
Semester:	5./6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Übungen / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie kennen die grundlegenden Konzepte der funktionalen Programmierung. Darüber hinaus sind Sie in der Lage, funktionale Programme in der Programmiersprache Haskell zu schreiben.
Inhalt:	<p>In der funktionalen Programmierung wird spezifiziert, <i>was</i> berechnet werden soll, nicht <i>wie</i> es berechnet werden soll. Spezifikation und Programm sind identisch, daher ist Korrektheit automatisch gegeben.</p> <p>Die Vorlesung bietet ein einiges an Theorie, aber immer in einer Form, die den Nutzen für die Praxis offenkundig werden lässt. In begleitenden Übungen werden die erlernten Konzepte praktisch angewendet und geübt und im Computerlabor umgesetzt. Hierbei lernen Sie die Programmiersprache Haskell.</p> <ol style="list-style-type: none"> 1. Programmiersprache Haskell 2. Listen 3. Rekursion, Iteration 4. Lambda-Ausdrücke 5. Funktionen höherer Ordnung 6. Lazy evaluation
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	

Informatik-Seminar

Modulbezeichnung:	Informatik-Seminar
Modulverantwortlicher:	Prof. Dr. H.W. Lang
Lehrender:	Prof. Dr. H.W. Lang
Semester:	6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach
Lehrform / SWS:	Seminar, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 30 h, Eigenstudium: 120 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie sind in der Lage, ein Thema selbstständig zu erarbeiten, didaktisch aufzubereiten und vor einem größeren Publikum vorzutragen.
Inhalt:	Die Informatik entwickelt sich schnell. Es gibt laufend neue Trends, Programmiersprachen, Frameworks usw. Sie arbeiten sich in ein solches Thema aus dem Bereich der neueren Entwicklungen der Informatik ein und stellen es im Informatik-Seminar vor. Ziel ist es, die Bedeutung und Anwendbarkeit dieser Neuentwicklungen kennenzulernen und einzuschätzen.
Prüfungsform:	SL: Sonstige Prüfungsleistung
Literatur:	

Interface- und Interaktionsdesign

Modulbezeichnung:	Interface- und Interaktionsdesign
Modulverantwortlicher:	Prof. Dr. S. Bertel
Lehrender:	Prof. Dr. S. Bertel
Semester:	1. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Usability und Mobile)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie kennen die grundlegenden Interface-Prinzipien und Interaktions-Muster und können damit umgehen. Sie sind in der Lage, Interfaces und Interaktionsabläufe zu entwickeln und kritisch zu beurteilen.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Geschichte des Interface-Designs • Usability-Engineering • Kognitive Grundlagen • Prinzipien • Metaphern • Interface-Design-Muster • Grafische Benutzer-Schnittstellen (GUI) • Fenstersysteme (WIMP)
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	<p>M. DAHM: Grundlagen der Mensch-Computer Interaktion. Pearson Studium (2006)</p> <p>S. KRUG: Don't make me think! Web Usability. 2. Auflage, Mitp (2006)</p> <p>B. SHNEIDERMAN, C. PLAISANT: Designing the User Interface. 4. Auflage, Pearson Education (2005)</p> <p>J. TIDWELL: Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly (2005)</p>

IT-Sicherheit

Modulbezeichnung:	IT-Sicherheit
Modulverantwortlicher:	Prof. Dr. S. Gajek
Lehrender:	Prof. Dr. S. Gajek
Semester:	5. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Computernetze und IT-Sicherheit)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	
Inhalt:	
Prüfungsform:	PL: Klausur
Literatur:	

Kommunikation und Präsentation

Modulbezeichnung:	Kommunikation und Präsentation
Modulverantwortlicher:	N.N.
Lehrender:	N.N.
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Übergreifende Qualifikationen)
Lehrform / SWS:	Workshop, 2 SWS
Arbeitsaufwand:	Präsenzstudium: 30 h, Eigenstudium: 45 h, Gesamtaufwand: 75 h
Leistungspunkte:	2.5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie können sich selbst und Ihre Vorhaben und Ziele klar, kompetent, dem Adressaten und der Situation angemessen darstellen.
Inhalt:	<p>Kommunikationstraining</p> <ul style="list-style-type: none"> • Kommunikationsmodelle • Kommunikationsprozesse in Gruppen • Fragetechniken • Kreativitätstechniken • Verkaufsgespräche und Einwandbehandlung • Vorstellungsgespräche und Gehaltsverhandlungen <p>Präsentationstraining</p> <ul style="list-style-type: none"> • Präsentationsanlässe, -inhalte und -formen • Präsentationstraining mit Videokamera • Vortrag • Präsentationen
Prüfungsform:	SL: Sonstige Prüfungsleistung
Literatur:	M. HARTMANN, R. FUNK, H. NIETMANN: Präsentieren. Beltz (2008)

Konzepte moderner Programmiersprachen

Modulbezeichnung:	Konzepte moderner Programmiersprachen
Modulverantwortlicher:	Prof. Dr. J. Christiansen
Lehrender:	Prof. Dr. J. Christiansen
Semester:	6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie finden sich schnell in einer Ihnen bisher unbekanntem Programmiersprache zurecht. Sie können anhand des Paradigmas der Programmiersprache einschätzen, welche Konzepte die Sprache zur Verfügung stellt. Sie kennen die Vor- und Nachteile verschiedener Konzepte in Hinsicht auf Aspekte des Software-Engineerings.
Inhalt:	<p>Sie lernen unterschiedliche Konzepte kennen, die von aktuellen Programmiersprachen unterstützt werden. Dazu werden die Programmiersprachen zuerst in verschiedene Klassen (Programmierparadigmen) eingeteilt. Anschließend werden die Konzepte vorgestellt, die von den jeweiligen Paradigmen unterstützt werden.</p> <ul style="list-style-type: none">• Paradigmen: imperativ, objekt-orientiert, funktional, logisch, multi-paradigmatisch• Typsysteme: statisch, dynamisch• Polymorphismus, Generics• Typinferenz• Funktionen höherer Ordnung, Lambda-Ausdrücke, Closures• Iteration, Rekursion, Iteratoren, Enumeratoren• Datenstrukturen: veränderbar, unveränderbar• Referenzen• Speichermanagement (Garbage-Collection, Reference-Counting) <p>In den Laborübungen erproben Sie die Umsetzung der verschiedenen Konzepte jeweils in einer Programmiersprache und vergleichen das Ergebnis mit der entsprechenden Umsetzung in anderen Sprachen.</p>
Prüfungsform:	PL: K(2) und HA
Literatur:	B.C. PIERCE: Types and Programming Languages. Mit Press (2002)

Kryptografie

Modulbezeichnung:	Kryptografie
Modulverantwortlicher:	Prof. Dr. H.W. Lang
Lehrender:	Prof. Dr. H.W. Lang
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Computernetze und IT-Sicherheit)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie kennen die mathematischen Grundlagen und die Funktionsweise der wichtigsten kryptografischen Verfahren für Verschlüsselung, Signatur und Authentifizierung. Sie kennen mögliche Angriffsmethoden und Sie sind in der Lage, die Sicherheit der kryptografischen Verfahren zu beurteilen.
Inhalt:	<p>Vorlesung</p> <p>Klassische Kryptografie</p> <ul style="list-style-type: none"> • Caesar Chiffre, Vernam Chiffre <p>Mathematische Grundlagen</p> <ul style="list-style-type: none"> • Teilbarkeit, Primzahl, Modulo-Rechnung • Gruppe, Ring, Körper <p>Public-Key-Verschlüsselung</p> <ul style="list-style-type: none"> • Diffie-Hellman-Schlüsselvereinbarung • RSA-Verfahren, ElGamal-Verfahren • Elliptische Kurven <p>Digitale Signatur</p> <ul style="list-style-type: none"> • RSA-Signatur, Kryptografische Hash-Funktionen <p>Zahlentheoretische Algorithmen</p> <ul style="list-style-type: none"> • Modulare Exponentiation • Erweiterter Euklidischer Algorithmus • Primzahltest <p>Kryptografische Protokolle</p> <ul style="list-style-type: none"> • Festlegung, Authentifizierung • Zero-Knowledge-Protokoll <p>Labor</p> <p>Im Computerlabor werden die verschiedenen zahlentheoretischen Algorithmen programmiert und darauf aufbauend das RSA-Verfahren implementiert. Programmiersprache ist Python.</p>
Prüfungsform:	PL: Klausur
Literatur:	<p>A. BEUTELSPACHER, H.B. NEUMANN, T. SCHWARZPAUL: Kryptografie in Theorie und Praxis. Vieweg (2005)</p> <p>J.A. BUCHMANN: Einführung in die Kryptographie. 4. Auflage, Springer (2008)</p> <p>C. KARPFIGER, H. KIECHLE: Kryptologie. Vieweg+Teubner (2010)</p>

Mathematik 1

Modulbezeichnung:	Mathematik 1
Modulverantwortlicher:	Prof. Dr. M. Kersken
Lehrender:	Prof. Dr. M. Kersken
Semester:	1. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Informatik-Grundlagen)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie beherrschen die für die Informatik wichtigen Grundlagen der Mengenlehre, der Algebra und der Geometrie. Sie können die hier vermittelten Methoden zur Modellierung und Lösung von Problemen anwenden. Sie werden an die abstrakte und axiomatische Denkweise der Algebra herangeführt.
Inhalt:	<p>Vorlesung</p> <p>Grundlagen</p> <ul style="list-style-type: none"> • Mengen, Relationen, Abbildungen, Logik, Boolesche Algebra • natürliche Zahlen und vollständige Induktion • ganze Zahlen und Restklassenarithmetik <p>Algebra</p> <ul style="list-style-type: none"> • algebraische Strukturen, Homomorphismen • Vektorräume, lineare Abbildungen • Matrizenrechnung, lineare Gleichungssysteme, Determinante <p>Analytische Geometrie</p> <ul style="list-style-type: none"> • lineare Geometrie • lineare Optimierung • quadratische Formen, Hauptachsentransformation, Kegelschnitte <p>Übungen</p> <p>1-std. in kleinen Gruppen</p>
Prüfungsform:	PL: Klausur
Literatur:	<p>W. PREUß, G. WENISCH, R.U. SCHMIDT: Lehr- und Übungsbuch Mathematik 1. 2. Auflage, Hanser (2003)</p> <p>P. STINGL: Mathematik für Fachhochschulen. 7. Auflage, Hanser (2003)</p> <p>U. STORCH, H. WIEBE: Lehrbuch der Mathematik 2. 2. Auflage, Spektrum (1999)</p>

Mathematik 2

Modulbezeichnung:	Mathematik 2
Modulverantwortlicher:	Prof. Dr. M. Kersken
Lehrender:	Prof. Dr. M. Kersken
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Informatik-Grundlagen)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie beherrschen die grundlegenden Themen der Analysis und der diskreten Mathematik über den allgemeinen Schulstoff hinaus. Nach der Bearbeitung themenbezogener Übungsaufgaben sind Sie mit den analytischen Methoden zur Modellierung und Lösung von Problemen vertraut.
Inhalt:	<p>Analysis</p> <ul style="list-style-type: none">• Folgen und Reihen• Differenzialrechnung einer Veränderlichen• Integralrechnung einer Veränderlichen• Reihenentwicklung von Funktionen• Numerische Lösungsverfahren von Gleichungen• Extremwertprobleme <p>Diskrete Mathematik</p> <ul style="list-style-type: none">• Kombinatorik• Diskrete Wahrscheinlichkeitsrechnung• Elemente der Graphentheorie, Netzwerke
Prüfungsform:	PL: Klausur
Literatur:	H. HEUSER: Lehrbuch der Analysis, Tl. 2. 13. Auflage, Teubner (2004) H. HEUSER: Lehrbuch der Analysis, Tl. 1. 16. Auflage, Teubner (2006) W. PREUß, G. WENISCH, R.U. SCHMIDT: Lehr- und Übungsbuch Mathematik 1. 2. Auflage, Hanser (2003) U. STORCH, H. WIEBE: Lehrbuch der Mathematik 2. 2. Auflage, Spektrum (1999)

Medizinische Visualisierung

Modulbezeichnung:	Medizinische Visualisierung
Modulverantwortlicher:	Prof. Dr. M. Teistler
Lehrender:	Prof. Dr. M. Teistler
Semester:	5./6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none">• Einführung in die verschiedenen Verfahren der medizinischen Bildgebung (Röntgen, Sonographie, Computertomographie, Kernspintomographie u.a.)• Klinische Bedeutung der medizinischen Bildgebung / wie arbeitet ein Radiologe?• Visualisierungstechniken• Medizinische Visualisierung und Informatik: ausgewählte Themen <p>Hausarbeit</p> <ul style="list-style-type: none">• Entwicklung und Präsentation von Konzepten und Softwareprototypen als Gruppenarbeit; hier sind verschiedene Themen möglich, auch eigene Vorschläge insbesondere aus den Bereichen Human-Computer-Interaction, Usability, 3D-Visualisierung• es gibt ggf. die Möglichkeit, die Hausarbeit mit einer Projektarbeit zu verknüpfen (Beispielprojekte aus dem Themenumfeld: http://www.userad.info)
Prüfungsform:	PL: Hausarbeit
Literatur:	

Mobile Computing

Modulbezeichnung:	Mobile Computing
Modulverantwortlicher:	Prof. Dr. T. Aschmoneit
Lehrender:	Prof. Dr. T. Aschmoneit
Semester:	5. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Usability und Mobile)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie kennen die wichtigsten aktuellen mobilen Betriebssysteme sowie Kommunikationsstandards. Sie können einfache Anwendungen für die iPhone/iPod-Plattform entwickeln.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Mobile Betriebssysteme • Schnittstellen und Kommunikationsstandards • Laufzeitumgebungen • Entwicklungsumgebungen • Plattformübergreifende Entwicklung • Vertriebswege • Synchronisationsprotokolle • Push/Poll-Anwendungen <p>Labor</p> <ul style="list-style-type: none"> • Mac OS X, iPhone Developer Program, Design Pattern • Objective C und Cocoa Touch • XCode, Interface Builder, Simulator • Anwendung 1: Views • Anwendung 2: Sensoren und GPS • Performance und Debugging
Prüfungsform:	PL: Klausur oder sonstige Prüfungsleistung
Literatur:	D. MARK, J. LEMARCHE: Beginning Iphone 3 Development: Exploring the Iphone SDK. Apress (2009) E. SADUN: The iPhone Developer's Cookbook: Building Applications with the iPhone 3.0 SDK. 2. Auflage, Addison-Wesley Longman (2010)

Netzwerk-Administration

Modulbezeichnung:	Netzwerk-Administration
Modulverantwortlicher:	Dipl.-Math. J. Stamp
Lehrender:	Dipl.-Math. J. Stamp
Semester:	3. Semester
Sprache:	deutsch / englisch
Zuordnung zum Curriculum:	Pflichtfach (Computernetze und IT-Sicherheit)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie beherrschen die Planung, den Aufbau und die Pflege von IP-basierten Computernetzen.
Inhalt:	<ul style="list-style-type: none"> • Networking Devices: Repeater, Hub, Transceiver, Bridge, Switch, Router, Gateway • <u>LAN, WAN, WLAN, VPN</u> • Protocol Stacks: <u>ISO/OSI, TCP/IP, IEEE802.x</u> and Real World • Encapsulation and Addressing • LAN-, Network and Transport Headers • Physical and Logical Addressing • IP Addressing: Reserved and Private Addresses • <u>IPv4 vs. IPv6</u> • IP Subnetting and Classless Routing • Internet Protocols and Applications: <u>Ethernet, IP, ICMP, ARP, DHCP, TCP, UDP, SMTP, SNMP, Telnet, FTP, DNS</u> • Static and Dynamic Routing • Access Control Lists and Firewalls
Prüfungsform:	PL: Klausur
Literatur:	<p>Cisco Networking Academy Course CCNA Exploration. netacad.fh-flensburg.de</p> <p>K.R. FALL, W.R. STEVENS: TCP/IP Illustrated Volume 1. 2. Auflage, Addison-Wesley (2011)</p> <p>D. LOWE: Networking All-in-One For Dummies. 4. Auflage, John Wiley & Sons (2010)</p> <p>W. RIGGERT: Rechnernetze. 3. Auflage, Hanser (2005)</p> <p>R. SCHREINER: Computernetzwerke. 3. Auflage, Hanser (2009)</p> <p>A.S. TANENBAUM: Computernetzwerke. 5. Auflage, Pearson Studium (2012)</p>

Objektorientierte Programmierung

Modulbezeichnung:	Objektorientierte Programmierung
Modulverantwortlicher:	Prof. Dr. M. Teistler
Lehrender:	Prof. Dr. M. Teistler
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Softwaretechnik)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	<p>Sie können einen objektorientierten Entwurf selbständig erstellen und in der Programmiersprache Java implementieren.</p> <p>Sie sind in der Lage, einfache lineare Datenstrukturen programmtechnisch umzusetzen, sowohl mithilfe der entsprechenden Collections-Klassen als auch mithilfe eigener Klassen.</p>
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none">• Konzept der Objektorientierung• Klassen und Objekte (Abstraktion und Kapselung, Aggregate, Initialisierung)• Vererbung (Ist- und Hat-Beziehung, Entwurf)• Polymorphie (statische und dynamische Bindung, abstrakte Klassen, Zweck, Anwendung)• Schnittstellen (Interfaces)• Generische Programmierung, Collections-Klassenbibliothek <p>Labor</p> <p>In den Laborübungen werden Programmieraufgaben zu o.g. Themen gestellt, die in der Sprache Java zu lösen sind. Die Lösungen werden analysiert und diskutiert.</p>
Prüfungsform:	PL: Klausur
Literatur:	

Plugin-Entwicklung mit Visual Studio

Modulbezeichnung:	Plugin-Entwicklung mit Visual Studio
Modulverantwortlicher:	O. Preikszas, B.Sc.
Lehrender:	O. Preikszas, B.Sc.
Semester:	5./6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie können eigene Plugins mithilfe der Entwicklungsumgebung Visual Studio entwickeln und testen.
Inhalt:	<ul style="list-style-type: none">• Architektur der Entwicklungsumgebung Visual Studio• Programmiersprache C#• Lebenszyklus eines Plugins• Implementierung eines Plugins• Debugging von Plugins• Testen von Plugins
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	W. DOBERENZ, T. GEWINNUS: Visual C# 2015. Hanser (2015)

Projekt

Modulbezeichnung:	Projekt
Modulverantwortlicher:	N.N.
Lehrender:	N.N.
Semester:	6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach
Lehrform / SWS:	Projekt, 12 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 390 h, Gesamtaufwand: 450 h
Leistungspunkte:	15 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie sind in der Lage, im Team ein anspruchsvolles Projekt aus dem Bereich der Web- und Softwaretechnologie durchzuführen. Sie wissen dabei die entsprechenden Techniken des Projektmanagements anzuwenden. Sie können die Ergebnisse Ihres Projekts in einer öffentlichen Präsentation attraktiv darstellen und in einem Abschlussbericht in verständlicher Form zusammenfassen.
Inhalt:	<p>Die Projekt-Arbeitsgruppe durchläuft folgende Stationen:</p> <p>Spezifizierung</p> <ul style="list-style-type: none"> • Aufgabenstellung • Projektdefinition <p>Strukturierung</p> <ul style="list-style-type: none"> • Arbeitspakete • Meilensteine • Terminplan <p>Umsetzung</p> <ul style="list-style-type: none"> • praktische Arbeit • Programmierung <p>Dokumentation</p> <ul style="list-style-type: none"> • Präsentation • Abschlussbericht
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	<p>H.-D. LITKE, I. KUNOW: Projektmanagement: Einfach! Praktisch!. 5. Auflage, Haufe (2006)</p> <p>H.-D. LITKE: Projektmanagement: Methoden, Techniken, Verhaltensweisen. Evolutionäres Projektmanagement. 5. Auflage, Hanser (2007)</p> <p>P. MANGOLD: IT-Projektmanagement kompakt. 2. Auflage, Spektrum (2004)</p>

Projektmanagement

Modulbezeichnung:	Projektmanagement
Modulverantwortlicher:	Prof. Dr. J. Christiansen
Lehrender:	Prof. Dr. J. Christiansen
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Übergreifende Qualifikationen)
Lehrform / SWS:	Workshop, 2 SWS
Arbeitsaufwand:	Präsenzstudium: 30 h, Eigenstudium: 45 h, Gesamtaufwand: 75 h
Leistungspunkte:	2.5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie beherrschen die wichtigsten inhaltlichen, theoretischen und operativen Techniken im Bereich des Projektmanagements. Sie können ein Projekt mittels geeigneter Instrumente planen, durchführen, präsentieren und dokumentieren. Sie kennen Instrumente und Tools aus dem Bereich des klassischen Projektmanagements sowie agile Methoden und Sie können beurteilen, wann welche Verfahren sinnvoll sind.
Inhalt:	<ul style="list-style-type: none">• Aufgaben des Projektmanagements• Projektarten• Projektphasen• Pflichtenheft• Lastenheft• Tools für das Projektmanagement• Rollen• agiles Projektmanagement• Scrum
Prüfungsform:	SL: Klausur oder sonstige Prüfungsleistung
Literatur:	M. COHN: Agile Estimating and Planning. Prentice Hall (2005) P. MANGOLD: IT-Projektmanagement kompakt. 2. Auflage, Spektrum (2004)

Recht

Modulbezeichnung:	Recht
Modulverantwortlicher:	N.N.
Lehrender:	N.N.
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Übergreifende Qualifikationen)
Lehrform / SWS:	Vorlesung, 2 SWS
Arbeitsaufwand:	Präsenzstudium: 30 h, Eigenstudium: 45 h, Gesamtaufwand: 75 h
Leistungspunkte:	2.5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie haben ein Grundverständnis für juristische Probleme, so dass Sie in ihrer späteren Tätigkeit die auftretenden rechtlichen Probleme angemessen beurteilen können. Sie sind in der Lage zu erkennen, wann ein rechtliches Problem von Ihnen gelöst werden kann und wann ein Berater hinzuziehen ist.
Inhalt:	<ul style="list-style-type: none">• Staatsorganisation, Grundrechte des Grundgesetzes mit internationalem Bezug zur Charta der Vereinten Nationen• Grundzüge des Öffentlichen Rechts und des Strafrechts• Grundzüge des Prozessrechts, insbesondere auch Mahnverfahren• Einführung in das Bürgerliche Recht (Allgemeiner Teil, Schuldrecht, Sachenrecht, Familien- und Erbrecht)• Vertragsgestaltung
Prüfungsform:	SL: Klausur oder sonstige Prüfungsleistung
Literatur:	E. KLUNZINGER: Einführung in das Bürgerliche Recht. 13. Auflage, Vahlen (2007)

Software Engineering 1

Modulbezeichnung:	Software Engineering 1
Modulverantwortliche:	Prof. Dr. M. Zachow
Lehrende:	Prof. Dr. M. Zachow
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Softwaretechnik)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie können Programme auf allen Ebenen der Testpyramide systematisch testen. Sie kennen verschiedene Software-Entwurfsmuster und können diese geeignet einsetzen. Sie können die Struktur und das Verhalten von Software mittels der UML-Notation spezifizieren und dokumentieren. Sie können Builds mit dem Build-Management-Tool Maven automatisieren.
Inhalt:	<p>Vorlesung</p> <p>Software-Engineering ist die Lehre von der ingenieurmäßigen Entwicklung, Erstellung und Wartung von großen Software-Paketen, unter Anwendung des jeweiligen Standes der Technik in den Teilbereichen Spezifikation, Architektur, Programmierung, Test und Dokumentation.</p> <ul style="list-style-type: none"> • Testmethoden für Software • Software/Architektur-Entwurfsmuster (Design Patterns) • UML-Diagramme • Versionsmanagement • Continuous Integration • Build Automatisierung <p>Übungen / Labor</p> <p>In begleitenden Laborübungen werden Beispiele am Computer programmiert.</p>
Prüfungsform:	PL: Klausur
Literatur:	<p>B. BRÜGGE, A.H. DUTOIT: Objektorientierte Softwaretechnik. Pearson Studium (2004)</p> <p>E. FREEMAN, E. FREEMAN: Head First Design Patterns. O'Reilly (2004)</p> <p>M. HITZ, G. KAPPEL, E. KAPSAMMER, W. RETSCHITZEGGER: UML@Work. 3. Auflage, dpunkt (2005)</p> <p>H. STÖRRLE: UML 2 für Studenten. Pearson Studium (2005)</p>

Software Engineering 2

Modulbezeichnung:	Software Engineering 2
Modulverantwortliche:	Prof. Dr. M. Zachow
Lehrende:	Prof. Dr. M. Zachow
Semester:	5. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Softwaretechnik)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	
Inhalt:	
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	

Strukturierte Programmierung

Modulbezeichnung:	Strukturierte Programmierung
Modulverantwortlicher:	Prof. Dr. J. Christiansen
Lehrender:	Prof. Dr. J. Christiansen
Semester:	1. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Softwaretechnik)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie beherrschen die grundlegenden Konzepte der Programmierung in einer höheren Programmiersprache. Sie können selbstständig einfache Problemlösungen formulieren und in der Programmiersprache Java implementieren.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none">• Syntax und Semantik• Variable, Wertzuweisung• Ein- und Ausgabe• Einfache Datentypen: int, double, char, boolean• Kontrollstrukturen: if..else, while, for• Funktionen: Definition, Aufruf, Parameterübergabe, Rückgabewert, Rekursion• Strukturierte Datentypen: Arrays, Strings• Exceptions• Abstrakte Datentypen; Klassen und Objekte <p>Labor</p> <p>Die Teilnahme am Labor ist verbindlich für den Laborschein. In den ersten Wochen wird langsam vorgegangen, damit auch Programmierneulinge die Möglichkeit haben, zu denjenigen, die schon einmal programmiert haben, aufzuschließen. Es wird dringend geraten, diese Phase entsprechend zu nutzen.</p>
Prüfungsform:	PL: Klausur
Literatur:	G. KRÜGER, T. STARK: Handbuch der Java-Programmierung. 6. Auflage, Addison-Wesley Longman (2009) H. MÖSSENBÖCK: Sprechen Sie Java?. 4. Auflage, dpunkt (2011) D. RATZ, J. SCHEFFLER, D. SEESE, J. WIESENBERGER: Grundkurs Programmieren in Java. 6. Auflage, Hanser (2011)

Systeme der Mobilkommunikation

Modulbezeichnung:	Systeme der Mobilkommunikation
Modulverantwortlicher:	Prof. Dr. T. Aschmoneit
Lehrender:	Prof. Dr. T. Aschmoneit
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Usability und Mobile)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	<p>Sie kennen die Grundlagen der digitalen Mobilkommunikation und die aktuell im Einsatz befindlichen Systeme, hier insbesondere die GSM- und LTE-Systeme mit ihren Erweiterungen.</p> <p>Sie kennen die Funktionsweisen und Anwendungsmöglichkeiten der weiteren Systeme wie etwa Bluetooth.</p>
Inhalt:	<ul style="list-style-type: none"> • Digitale Modulation • Übersicht Mobilfunkkanal • Zellulare Netze, Netzplanung • GSM Global System for Mobilcommunications • Systemarchitektur • Protokolle • Evolution von GSM zu <u>LTE</u>; <u>EDGE</u>, <u>HSCSD</u>, <u>GPRS</u>, <u>UMTS</u>, <u>LTE</u> • Anwendungsaspekte und Application Interfaces <p>Weitere Systeme im Überblick:</p> <ul style="list-style-type: none"> • <u>DECT</u> • Bluetooth • <u>TETRA</u> • Satellite Systems • <u>GPS</u>, INMARSAT, Internet via Satellit
Prüfungsform:	PL: Klausur oder sonstige Prüfungsleistung
Literatur:	<p>J. EBERSPÄCHER, H.J. VÖGEL, C. BETTSTETTER: GSM Global System for Mobile Communication. Teubner (2001)</p> <p>H. KAARANEN, A. AHTIAINEN, L. LAITINEN: UMTS Networks. 2. Auflage, Wiley & Sons (2005)</p> <p>A. MERKLE, A. TERZIS: Digitale Funkkommunikation mit Bluetooth. Franzis (2002)</p> <p>M. SAUTER: Grundkurs Mobile Kommunikationssysteme. 5. Auflage, Springer Vieweg (2013)</p> <p>B. WALKE: Mobilfunknetze und ihre Protokolle, Bd. 1. 3. Auflage, Teubner (2001)</p> <p>B. WALKE: Mobilfunknetze und ihre Protokolle, Bd. 2. 3. Auflage, Teubner (2001)</p>

Systemprogrammierung

Modulbezeichnung:	Systemprogrammierung
Modulverantwortlicher:	Prof. Dr. W. Tepper
Lehrender:	Prof. Dr. W. Tepper
Semester:	5./6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie können ein Problem in parallel ausführbare Einheiten zerlegen und mithilfe systemnaher Programmierung effizient umsetzen. Dabei sind Sie in der Lage, die Mechanismen der Synchronisation und Kommunikation für die Kooperation nebenläufiger Programmteile zu nutzen.
Inhalt:	<ul style="list-style-type: none">• Programmieren in C• Nebenläufigkeit (Prozesse, Threads)• Synchronisation (Kooperation und gegenseitiger Ausschluss, Semaphore- und Monitorkonzept, Verklemmungen)• Kommunikation (Signale, gemeinsamer Speicher, Pipes)
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	

Theoretische Informatik

Modulbezeichnung:	Theoretische Informatik
Modulverantwortliche:	Dr. P. Sadeghi
Lehrende:	Dr. P. Sadeghi
Semester:	5. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Informatik-Grundlagen)
Lehrform / SWS:	Vorlesung / Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie kennen die wichtigsten theoretischen Konzepte aus dem Bereich der Formalen Sprachen und Automaten sowie der Komplexitätstheorie. Sie können auf Basis einer Grammatik einen Parser und Übersetzer bauen.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none"> • Alphabet, Wort, Sprache, Grammatik • Regulärer Ausdruck, reguläre Sprache • Nichtdeterministischer endlicher Automat • Kontextfreie Sprachen und Stack-Automaten • Recursive-Descent-Parser und -Übersetzer • Turing-Maschine, Berechenbarkeit • Komplexitätstheorie <p>Übungen / Labor</p> <p>In der ersten Semesterhälfte werden wöchentlich Übungsaufgaben gestellt, die korrigiert werden und in gemeinsamen Übungsstunden besprochen werden.</p> <p>In der zweiten Semesterhälfte wird in Laborübungen ein Recursive-Descent-Übersetzer für arithmetische Ausdrücke gebaut.</p> <p>Die Programmiersprache ist Java oder Python.</p>
Prüfungsform:	PL: Mündliche Prüfung
Literatur:	<p>A. ASTEROTH, C. BAIER: Theoretische Informatik. Pearson Studium (2002)</p> <p>D.W. HOFFMANN: Theoretische Informatik. Hanser (2009)</p> <p>B. HOLLAS: Grundkurs Theoretische Informatik. Spektrum Akademischer Verlag (2007)</p> <p>U. SCHÖNING: Theoretische Informatik kurz gefasst. BI-Wissenschaftsverlag (1992)</p> <p>M. SIPSER: Introduction to the Theory of Computation. PWS Publishing Company (1996)</p>

Usability Engineering

Modulbezeichnung:	Usability Engineering
Modulverantwortlicher:	Prof. Dr. S. Bertel
Lehrender:	Prof. Dr. S. Bertel
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Usability und Mobile)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie kennen gängige Testmethoden, Testwerkzeuge und Testverfahren im Bereich Usability. Sie haben die Verfahren praktisch angewendet und können für verschiedene Anwendungen Usability-Tests konzipieren, planen und durchführen.
Inhalt:	<p>Vorlesung</p> <ul style="list-style-type: none">• Theorie und Grundbegriffe (ISO-Norm, Einflussfaktoren und Ziele)• Usability im Software-Entwicklungsprozess (Zielgruppenanalyse, Personas, Prototyping, Informationsarchitektur, Szenarios)• Usability Testing (Thinking Aloud, Kontextanalyse, Eyetracking, Panelbefragung, Crowd-Testing, A/B Tests)• Usability-Metriken und -Richtlinien• Mobile Usability <p>Labor</p> <p>In den begleitenden Laboren konzipieren und planen Sie Usability-Tests für bestehende Software, führen die Tests im Usability-Labor durch und werten sie anschließend aus.</p>
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	

Verteilte Webanwendungen und Webservices

Modulbezeichnung:	Verteilte Webanwendungen und Webservices
Modulverantwortlicher:	B. Schulz, B.Sc.
Lehrender:	B. Schulz, B.Sc.
Semester:	5./6. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Wahlpflichtfach
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie können verteilte service-orientierte Anwendungen entwerfen und auf Basis des Microsoft-Technologie-Stacks realisieren. Darüber hinaus sind Sie in der Lage, aktuelle fortgeschrittene Programmier Techniken anzuwenden und zu bewerten.
Inhalt:	Serverseitige Programmierung <ul style="list-style-type: none">• Web-Anwendungen und REST-Webservices mit <u>MVC</u>• Webservices und die Windows Communication Foundation Fortgeschrittene Programmier Techniken <ul style="list-style-type: none">• funktionale Programmierung in C# (LINQ / Lambdas / higher-order functions)• Inversion of Control• Dependency Injection• Reflections• Composition via <u>MEF</u>
Prüfungsform:	PL: Hausarbeit
Literatur:	

Web-Programmierung

Modulbezeichnung:	Web-Programmierung
Modulverantwortlicher:	Prof. Dr. W. Tepper
Lehrender:	Prof. Dr. W. Tepper
Semester:	3. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Webtechnologien)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie beherrschen die client- und serverseitige Web-Programmierung. Sie können die Zustände einer Web-Anwendung analysieren und mit entsprechenden Konstruktionen realisieren. Sie haben erste Erfahrungen in Entwurf, Entwicklung und Test einer vollständigen Anwendung mit RIA-Clients, Server-Middleware und Datenbank gesammelt.
Inhalt:	<p>Vorlesung</p> <p>Serverseitige Programmierung</p> <ul style="list-style-type: none">• <u>C</u>G<u>I</u>• <u>P</u>H<u>P</u>• <u>A</u>S<u>P</u>.<u>N</u>E<u>T</u>• Node.js <p>Zustandsverwaltung</p> <ul style="list-style-type: none">• Ansicht• Sitzung• Anwendung <p>Programmierung von Webapplikationen</p> <ul style="list-style-type: none">• Persistenz mit NoSQL-Web-Datenbanken• <u>A</u>j<u>a</u>x• <u>H</u>T<u>M</u>L5 Communication <u>A</u>P<u>I</u>s• <u>R</u>I<u>A</u>-Clients <p>Übungen</p> <p>In den Übungen werden Programmieraufgaben zu o.g. Themen gestellt. Die Lösungen werden analysiert und diskutiert.</p>
Prüfungsform:	PL: Klausur oder sonstige Prüfungsleistung
Literatur:	WEB: PHP. http://php.net/docs.php WEB: ASP. http://www.asp.net/ S. EDLICH, A. FRIEDLAND, J. HAMPE, B. BRAUER, M. BRÜCKNER: NoSQL. 2. Auflage, Hanser (2011)

Web-Systeme

Modulbezeichnung:	Web-Systeme
Modulverantwortlicher:	Prof. Dr. W. Tepper
Lehrender:	Prof. Dr. W. Tepper
Semester:	4. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Webtechnologien)
Lehrform / SWS:	Workshop, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	Orientierungsprüfung
Lernziele / Kompetenzen:	Sie können die Anforderungen an ein komplexes webbasiertes System analysieren. Sie sind in der Lage, im Team über die zu verwendenden Sprachen, Werkzeuge und Frameworks zu entscheiden und damit das System erfolgreich zu realisieren.
Inhalt:	<p>Im Workshop werden wöchentlich neue weiterführende Themen erarbeitet und in einem Team praktisch erprobt und geübt. Die Themen sind zum Beispiel:</p> <ul style="list-style-type: none">• Web-Frameworks und -Libraries• Content-Management-Systeme• Multi-Browser-Support• Responsive Webdesign• Website Performance, Sharding• Externe Dienste, Mashups, Content Syndication• 2D/3D-Webgrafik• RESTful Services• Platzierung und Abrechnung von Werbeinhalten
Prüfungsform:	PL: Sonstige Prüfungsleistung
Literatur:	WEB: HTML5. http://html5boilerplate.com/ WEB: Sharding. http://www.codefutures.com/database-sharding/ WEB: Drupal. http://drupal.org/documentation WEB: jQuery. http://learn.jquery.com/

Webdesign

Modulbezeichnung:	Webdesign
Modulverantwortlicher:	Prof. Dr. W. Tepper
Lehrender:	Prof. Dr. W. Tepper
Semester:	1. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Webtechnologien)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Die grundlegenden Prinzipien und Fachbegriffe des Webs sind Ihnen bekannt. Sie beherrschen den Umgang mit Web-Entwicklungswerkzeugen. Sie können Webseiten anwendungsorientiert entwerfen und in HTML/CSS-Code umsetzen.
Inhalt:	<ul style="list-style-type: none">• Konzepte des World Wide Web (Client/Servermodell, HTTP, URI)• Entwurfsrichtlinien (Layout, Navigation, Farben, Typografie, Grafikformate)• Markup und Style mit HTML und CSS• Barrierefreiheit und Geräteabhängigkeit• Entwicklungswerkzeuge
Prüfungsform:	SL: Sonstige Prüfungsleistung
Literatur:	J.E. HELLBUSCH, K. PROBIESCH: Barrierefreiheit verstehen und umsetzen. dpunkt (2011) P. KRÖNER: HTML5. Webseiten innovativ und zukunftssicher. 2. Auflage, Open Source Press (2011) K. LABORENZ: CSS. Galileo Computing (2011)

Webtechnologien

Modulbezeichnung:	Webtechnologien
Modulverantwortlicher:	Prof. Dr. W. Tepper
Lehrender:	Prof. Dr. W. Tepper
Semester:	2. Semester
Sprache:	deutsch
Zuordnung zum Curriculum:	Pflichtfach (Webtechnologien)
Lehrform / SWS:	Vorlesung / Labor, 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 h, Eigenstudium: 90 h, Gesamtaufwand: 150 h
Leistungspunkte:	5 cp
Voraussetzungen:	keine
Lernziele / Kompetenzen:	Sie sind in der Lage, Webclient-Programme zu erstellen. Hierbei nutzen Sie Ihre Kenntnisse in XML oder JSON, um Informationen zu strukturieren. Sie können die in der Webentwicklung gebräuchlichen XML-Sprachen einschätzen und anwenden. Die Nutzung von HTML5-APIs ist Ihnen geläufig.
Inhalt:	<ul style="list-style-type: none">• Clientseitige Programmierung mit JavaScript• <u>DOM</u>-Scripting• Informationen strukturieren mit <u>XML</u> und <u>JSON</u>• XML-basierte Sprachen im Web: z.B. <u>RSS</u>, <u>SVG</u>, <u>MathML</u>• <u>HTML5</u> und JavaScript: z.B. Drag and Drop, Data Storage, Audio/Video, Geolocation, Offline Web
Prüfungsform:	PL: Klausur oder sonstige Prüfungsleistung
Literatur:	WEB: HTML5 APIs. http://www.w3.org/TR/2011/WD-html5-20110525/ D. FLANAGAN: JavaScript - Das umfassende Referenzwerk. 6. Auflage, O'Reilly (2012) J. KEITH, J. SAMBELLS: DOM Scripting. eBook / Friends of Ed (2010)