

**Prüfungsordnung
für den Masterstudiengang
Informatik
an der Fachhochschule Bielefeld
(University of Applied Sciences)
vom 17.06.2013**

in der Fassung der Änderung vom 19.05.2014 und 08.07.2015

Aufgrund des § 2 Abs. 4 und des § 64 Abs. 1 des Gesetzes über die Hochschulen des Landes Nordrhein-Westfalen (Hochschulgesetz - HG) vom 31. Oktober 2006 (GV. NRW. S. 474), zuletzt geändert durch Artikel 1 des Gesetzes vom 3. Dezember 2013 (GV. NRW. S. 723), hat der Fachbereich Technik der Fachhochschule Bielefeld die folgende Ordnung erlassen:

Inhaltsübersicht

I. Allgemeines	3
§ 1 Geltungsbereich der Prüfungsordnung.....	3
§ 2 Ziel des Studiums, Zweck der Prüfung, Akademischer Grad.....	3
§ 3 Allgemeine Zulassungsvoraussetzungen.....	3
§ 4 Regelstudienzeit, Studienumfang.....	4
§ 5 Arten des Lehrangebots.....	4
§ 6 Umfang und Gliederung der Prüfungen.....	5
§ 7 Organisation der Prüfungen, Prüfungsorgane.....	5
§ 8 Prüfende und Beisitzende.....	6
§ 9 Anrechnung von Studienzeiten, Studien- und Prüfungsleistungen.....	6
§ 10 Wiederholung von Prüfungsleistungen.....	7
§ 11 Versäumnis, Rücktritt, Täuschung, Ordnungsverstoß.....	7
II. Prüfungsabläufe	7
§ 12 Ziel, Umfang und Form der Modulprüfungen.....	7
§ 13 Zulassung zu Modulprüfungen.....	8
§ 14 Durchführung von Modulprüfungen.....	8
§ 15 Klausurarbeiten und sonstige schriftliche Arbeiten.....	9
§ 16 Mündliche Prüfungen.....	9
§ 17 Hausarbeiten.....	10
§ 18 Projektarbeiten.....	10
§ 19 Performanzprüfungen.....	10
§ 20 Abzuleistende Modulprüfungen, Credits.....	11
§ 21 Bewertung von Prüfungsleistungen.....	11
III. Masterarbeit (Thesis) und Kolloquium	12
§ 22 Masterarbeit (Thesis).....	12
§ 23 Zulassung zur Masterarbeit.....	12
§ 24 Ausgabe und Bearbeitung der Masterarbeit.....	12
§ 25 Abgabe und Bewertung der Masterarbeit.....	13
§ 26 Kolloquium.....	13
IV. Ergebnis der Masterprüfung, Zusatzmodule	14
§ 27 Ergebnis der Masterprüfung.....	14

§ 28	Zeugnis, Gesamtnote, Masterurkunde, Diploma Supplement.....	14
§ 29	Zusatzmodule	14
V. Schlussbestimmungen	15
§ 30	Einsicht in die Prüfungsakte	15
§ 31	Ungültigkeit von Prüfungen.....	15
§ 32	In-Kraft-Treten, Veröffentlichung	15
Anlage 1:	Studienverlaufsplan	16
Anlage 2:	Modulhandbuch	18

I. Allgemeines

§ 1

Geltungsbereich der Prüfungsordnung

Diese Masterordnung gilt für den Abschluss des Studiums in dem Masterstudiengang Informatik an der Fachhochschule Bielefeld. Sie regelt die Prüfungen in diesem Studiengang, sowie den Inhalt und Aufbau des Studiums unter Berücksichtigung der fachlichen und hochschuldidaktischen Entwicklungen und Anforderungen der beruflichen Praxis und enthält die inhaltliche Beschreibung der Prüfungsgebiete.

§ 2

Ziel des Studiums, Zweck der Prüfung, Akademischer Grad

- (1) Die Masterprüfung bildet den zweiten berufsqualifizierenden Abschluss eines Hochschulstudiums.
- (2) Das Masterstudium gewährleistet auf der Grundlage wissenschaftlicher Erkenntnisse und Methoden und unter Beachtung der allgemeinen gesetzlichen Studienziele (§ 58 HG) eine deutliche Berufsqualifizierung. Der Studiengang vermittelt daher den Absolventinnen und Absolventen Qualifikationsbündel bzw. -attribute, die ihnen die Aufnahme einer qualifikationsadäquaten beruflichen Tätigkeit nach dem Studium ermöglichen.
- (3) Im Rahmen des Pflicht- oder Wahlpflichtbereiches sind unter Beachtung der Maßgaben des Absatzes 2 folgende überfachliche Qualifikationen zu gewährleisten:
 1. Fähigkeit zu wissenschaftlichem Arbeiten einschließlich der dazu erforderlichen Informations- und Medienkompetenz;
 2. Fähigkeit zur Problemanalyse und der Strukturierung und Planung des Lösungsablaufs;
 3. Fähigkeit, Ideen, Konzepte, Projekte oder Produkte in mündlicher, schriftlicher und digitaler Form zu präsentieren;
 4. Fähigkeit, vor dem Hintergrund wissenschaftlicher Erkenntnisse und Methoden konkrete Fragestellungen des Berufsfeldes in einem vorgegebenen Zeitrahmen zu bearbeiten;
 5. Fähigkeit zur Teamarbeit und zur Übernahme von Leitungsaufgaben;
- (4) Aufgrund der bestandenen Masterprüfung wird der akademische Grad „Master of Science“ (M.Sc.) verliehen.

§ 3

Allgemeine Zulassungsvoraussetzungen

- (1) Voraussetzung für die Zulassung zum Studium ist der Nachweis eines abgeschlossenen Hochschulstudiums mit mindestens dem Abschluss Bachelor in einer betriebswirtschaftlichen, naturwissenschaftlichen, ingenieurwissenschaftlichen oder mathematischen Richtung mit guten Kenntnissen der angewandten Informatik sowie ausreichenden englischen Sprachkenntnissen.
- (2) Die Mindestanzahl der Credit Points des für den Masterstudiengang qualifizierenden Bachelorstudiengangs beträgt 210 Punkte. Dies entspricht in der Regel einem siebensemestrigen Bachelorstudiengang oder einem FH-Diplom. Mindestens 100 dieser Credit Points sollen in Fächern mit Informatikrelevanz erworben worden sein.
- (3) Hat eine Bewerberin oder ein Bewerber einen Abschluss mit nur 180 Credit Points - dies entspricht in der Regel einem sechssemestrigen Bachelorstudiengang - oder weniger als 100 Credit Points mit Informatikrelevanz, so legt der Prüfungsausschuss fest, wie die fehlenden Credit Points erworben werden können. Dies kann durch das erfolgreiche Belegen von Kursen im Bachelorstudiengang Informatik oder/und durch Absolvieren von Praktika in Unternehmen erfolgen. In Ausnahmefällen kann die Hochschule zulassen, dass das Studium vor Erwerb der Zulassungsvoraussetzungen nach den Sätzen 1 und 2 aufgenommen wird, wenn diese Zulassungsvoraussetzungen spätestens innerhalb eines Jahres nach Aufnahme des Studiums nachgewiesen werden.
- (4) Bei der Bewerbung sind folgende Unterlagen einzureichen:
 - Das Abschlusszeugnis des für den Masterstudiengang qualifizierenden Hochschulabschlusses und das dazugehörige Dokument (Transcript, o.ä.), das Auskunft gibt über den individuellen Studienverlauf, die besuchten Lehrveranstaltungen und Module, die in diesem Studium erbrachten Leistungen und deren Bewertungen sowie über das individuelle fachliche Profil des absolvierten Studiengangs. Falls die Hochschule, an der die Bewerberin oder der Bewerber den für den Masterstudiengang qualifizierenden

Hochschulabschluss erworben hat, kein entsprechendes Dokument ausfertigen kann, sind stattdessen die erworbenen Leistungsnachweise einzureichen.

- Ein ca. drei Seiten langes Schreiben in deutscher Sprache, das Aufschluss über die Motivation und Eignung der Bewerberin/des Bewerbers für diesen Masterstudiengang gibt.
- (5) Für das Studium sind gemäß Abs. 1 ausreichende Englisch-Kenntnisse nötig. Diese werden in der Regel in einem Bachelor-Studiengang erworben. Liegen keine ausreichenden Englisch-Kenntnisse vor, so sind diese zu erwerben und vor Beginn der Masterarbeit nachzuweisen.
 - (6) Sind mehr Bewerber/innen als Studienplätze vorhanden, so erstellt ein vom Prüfungsausschuss eingesetztes Auswahlgremium eine Rangliste hinsichtlich der weiteren Zulassung. Dieses Auswahlgremium besteht aus zwei Mitgliedern der Gruppe der Hochschullehrerinnen und Hochschullehrer, die im Masterstudiengang lehren, einem Mitglied aus der Gruppe der akademischen Mitarbeiterinnen und Mitarbeiter und einer/einem Studierenden als beratendes Mitglied.
 - (7) Eine Ablehnung des Zulassungsantrages schließt eine erneute Bewerbung zu einem späteren Termin nicht aus.
 - (8) Trotz des Vorliegens der allgemeinen Zulassungsvoraussetzungen kann die Einschreibung bzw. der Studiengangwechsel versagt werden, wenn die Studienbewerberin oder der Studienbewerber an einer Hochschule im Geltungsbereich des Grundgesetzes eine nach der Prüfungsordnung erforderliche Prüfung in einem verwandten oder vergleichbaren Studiengang endgültig nicht bestanden hat.

§ 4

Regelstudienzeit, Studienumfang

- (1) Die Erst-Immatrikulation erfolgt zum Beginn eines Winter- oder Sommersemesters. Die Regelstudienzeit beträgt anderthalb Jahre.
- (2) Der Studiengang ist modular aufgebaut. Jedes Modul schließt mit einer Prüfung ab. Der für ein Modul aufzuwendende Arbeitsaufwand wird durch Leistungspunkte (Credit Points) beschrieben. Credits umfassen sowohl die Lehrveranstaltungen als auch Zeiten für die Vor- und Nachbereitung des Lehrstoffes, die Prüfungsvorbereitungen einschließlich der Abschluss- und Studienarbeiten. Nach bestandener Prüfung werden die entsprechenden Leistungspunkte gutgeschrieben und getrennt von den erzielten Prüfungsnoten ausgewiesen. Entsprechend dem European Credit Transfer System (ECTS – Europäisches System zur Anrechnung von Studienleistungen) werden pro Semester 30 Credits vergeben und den Modulen zugeordnet.
- (3) Der Studienverlaufsplan (Anlage 1) legt den Arbeitsaufwand in Credits und den Zeitumfang der einzelnen Module, in Semesterwochenstunden und Credits sowie deren Art und empfohlene Zeitlage im Studiengang fest. Die Lehrveranstaltungen werden gewöhnlich im Jahresrhythmus angeboten. Das erste und zweite Semester können in der Reihenfolge frei belegt werden. Die spezifischen Prüfungsanforderungen, die Pflichtmodule und die Wahlpflichtmodule sind im Studienverlaufsplan (Anlage 1) verbindlich geregelt; dieses gilt auch für die Reihenfolge der abzuleistenden Module, soweit dies notwendig oder zweckmäßig ist.
- (4) Der Leistungsumfang beträgt in diesem dreisemestrigen Studiengang 90 Credits.
- (5) Im ersten Semester wird in einem Gespräch mit jeder/jedem Studierenden festgelegt, ob und welche Angleichungskurse zu belegen sind, ob ausreichende Englischkenntnisse nachgewiesen werden können und ob bereits 210 Credits im vorausgegangenen Studium erbracht wurden. Im Ergebnis des Gesprächs werden Maßnahmen zum Erreichen dieser Studienvoraussetzungen und zur verpflichtenden Teilnahme an Angleichungskursen festgelegt. Die Ausgabe eines Masterarbeitsthemas darf erst erfolgen, wenn alle Auflagen erfüllt und nachgewiesen worden sind.

§ 5

Arten des Lehrangebots

- (1) Das notwendige Lehrangebot enthält Pflicht- und Wahlpflichtmodule (siehe Anlage 1).
- (2) Wahlpflichtmodule sind Module, die als Prüfungsmodule gewählt und mit einer Modulprüfung abgeschlossen werden müssen.
- (3) Zusatzmodule sind freiwillig erbrachte Leistungen, für deren Anerkennung sich die Studierenden einer Prüfung (§ 12 MPO) unterziehen müssen.
- (4) Formen der Lehrveranstaltungen sind:
Vorlesung (V):
Zusammenhängende Darstellung eines Lehrstoffes, Vermittlung von Fakten und Methoden.

Seminaristischer Unterricht (SU):

Didaktischer Mix mit abwechselnden Phasen der Gruppenarbeit, Stoffvermittlung als Referat oder Lehrgespräch durch die Dozentin/den Dozenten und Kurzreferate der Studierenden.

Übung (Ü):

Systematisches Durcharbeiten von Lehrstoffen und Zusammenhängen, Anwendung auf Fälle aus der Praxis. Die Lehrenden leiten die Veranstaltungen, geben eine Einführung, stellen Aufgaben, geben Lösungshilfen. Die Studierenden arbeiten einzeln oder in Gruppen mit, lösen Aufgaben teilweise selbständig, aber in enger Rückkopplung mit den Lehrenden.

Praktikum (P):

Erwerben und Vertiefen von Kenntnissen durch Bearbeitung und Programmierung praktischer und experimenteller Aufgaben im IT-Lehrraum. Die Lehrenden leiten die Studierenden an und überwachen die Veranstaltung. Die Studierenden führen praktische Arbeiten und Versuche durch und erhalten darüber ein Testat.

§ 6

Umfang und Gliederung der Prüfungen

- (1) Das Studium wird mit der Masterprüfung abgeschlossen. Die Masterprüfung gliedert sich in studienbegleitende Modulprüfungen und Masterarbeit. Die studienbegleitenden Modulprüfungen sollen zu dem Zeitpunkt stattfinden, an dem das jeweilige Modul im Studium abgeschlossen wird.
- (2) Die Meldung zur Masterarbeit (Antrag auf Zulassung) soll in der Regel nach Abschluss des zweiten Semesters erfolgen.
- (3) Hinsichtlich der Leistungen und der zeitlichen Bestimmungen im Zusammenhang mit der Masterarbeit gelten die Regelungen gemäß §§ 22 - 26.
- (4) Das Studium sowie das Prüfungsverfahren sind so zu gestalten, dass einschließlich der Masterprüfung das Studium mit Ablauf des dritten Semesters abgeschlossen sein kann. Die Prüfungsverfahren müssen die Inanspruchnahme von Schutzbestimmungen entsprechend den §§ 3, 4, 6 und 8 des Mutterschutzgesetzes sowie entsprechend den Fristen des Bundeserziehungsgeldgesetzes über die Elternzeit und die Ausfallzeiten durch die Pflege von Personen nach § 48 Abs. 5 HG berücksichtigen (§ 64 Abs. 2 Nr. 5 HG).

§ 7

Organisation der Prüfungen, Prüfungsorgane

- (1) Für die Prüfungsorganisation ist die Dekanin oder der Dekan gemäß § 27 Abs. 1 Satz 2 HG verantwortlich. Diese Aufgaben können durch einen Prüfungsausschuss wahrgenommen werden.
- (2) Die Dekanin oder der Dekan oder der Prüfungsausschuss fungieren entsprechend ihrer Bestimmung in der Prüfungsordnung als Behörde im Sinne des Verwaltungsverfahrensgesetzes NRW und der Verwaltungsgerichtsordnung.
- (3) Wenn ein Prüfungsausschuss als Prüfungsbehörde eingerichtet wird, sollen in der Regel diesem Gremium nicht mehr als sieben Mitglieder angehören. In diesem Fall entspricht folgende Zusammensetzung den Maßgaben des HG:
 1. vier Mitglieder der Professorenschaft, darunter ein vorsitzendes Mitglied und ein stellvertretend vorsitzendes Mitglied,
 2. ein Mitglied der Mitarbeiterschaft in Lehre und Forschung mit Hochschulabschluss,
 3. zwei Studierende.
- (4) Die Mitglieder des Prüfungsausschusses werden vom zuständigen Fachbereichsrat gewählt. Entsprechend wird durch die Wahl bestimmt, wer die Mitglieder sind, mit Ausnahme des vorsitzenden Mitglieds und des stellvertretend vorsitzenden Mitglieds, welches das vorsitzende Mitglied im Verhinderungsfall vertreten soll. Die Amtszeit der Mitglieder beträgt vier Jahre, die eines studentischen Mitglieds ein Jahr. Die Wiederwahl eines Mitglieds ist möglich. Scheidet ein Mitglied vorzeitig aus, wird ein Nachfolger für die restliche Amtszeit gewählt.
- (5) Der Prüfungsausschuss achtet auf die Einhaltung der Prüfungsordnung. Er entscheidet insbesondere über Widersprüche gegen in Prüfungsverfahren getroffene Entscheidungen. Darüber hinaus hat der Prüfungsausschuss dem Fachbereichsrat über die Entwicklung der Prüfungen und Studienzeiten jährlich zu berichten. Er gibt Anregungen zur Reform der Prüfungsordnung. Der Prüfungsausschuss kann die Erledigung seiner Aufgaben für alle Regelfälle auf das vorsitzende Mitglied, bzw. das stellvertretend vorsitzende Mitglied des Prüfungsausschusses übertragen; dies gilt nicht für die Entscheidung über Widersprüche.
- (6) Der Prüfungsausschuss ist beschlussfähig, wenn das vorsitzende Mitglied (oder Stellvertretung), ein weiteres Mitglied der Professorenschaft und ein weiteres stimmberechtigtes

Mitglied anwesend ist. Er beschließt mit einfacher Stimmenmehrheit. Bei Stimmgleichheit entscheidet die Stimme des vorsitzenden Mitglieds. Die studentischen Mitglieder wirken bei pädagogisch-wissenschaftlichen Entscheidungen, insbesondere bei der Anrechnung oder sonstigen Beurteilung von Studien- und Prüfungsleistungen und der Bestellung von Prüfenden und Beisitzenden, nicht mit. An der Beratung und Beschlussfassung über Angelegenheiten, welche die Festlegung von Prüfungsaufgaben oder ihre eigene Prüfung betreffen, nehmen die studentischen Mitglieder des Prüfungsausschusses nicht teil.

- (7) Die Mitglieder des Prüfungsausschusses, mit Ausnahme der studentischen Mitglieder, die sich im gleichen Prüfungszeitraum der gleichen Prüfung unterziehen, haben das Recht, der Abnahme der Prüfungen beizuwohnen. Dieses Recht erstreckt sich nicht auf die Bekanntgabe der Note.
- (8) Die Mitglieder des Prüfungsausschusses (einschl. der Stellvertretung), die Prüfenden und die Beisitzenden unterliegen der Amtsverschwiegenheit. Sofern sie nicht im öffentlichen Dienst stehen, sind sie durch den Prüfungsausschuss zur Verschwiegenheit zu verpflichten.
- (9) Belastende Entscheidungen des Prüfungsausschusses sind dem betroffenen Studierenden unverzüglich schriftlich mitzuteilen. Der Bescheid ist mit einer Rechtsbehelfsbelehrung zu versehen.
- (10) Wenn die Prüfungsordnung die Dekanin oder den Dekan zur Prüfungsbehörde bestimmt, wird sie oder er im Falle der Verhinderung durch die Prodekanin oder den Prodekan vertreten. In der Tätigkeit als Prüfungsorgan wird die Dekanin oder der Dekan durch die Hochschulverwaltung unterstützt (§ 25 HG). Hinsichtlich der weiteren Aufgabenbestimmung gilt Abs. 6 sinngemäß.

§ 8

Prüfende und Beisitzende

- (1) Der Prüfungsausschuss bestellt die Prüfenden und Beisitzenden. Zum Prüfenden darf nur bestellt werden, wer mindestens die entsprechende Masterprüfung an einer Hochschule oder eine vergleichbare Prüfung abgelegt hat oder eine vergleichbare Qualifikation erworben hat und, sofern nicht zwingende Gründe eine Abweichung erfordern, in dem Studienabschnitt, auf den sich die Prüfung bezieht, eine einschlägige selbständige Lehrtätigkeit ausgeübt hat. Sind mehrere Prüfer zu bestellen, so soll mindestens eine prüfende Person in dem betreffenden Prüfungsfach gelehrt haben. Zu Beisitzenden dürfen nur Personen bestellt werden, die mindestens die Masterprüfung an einer Hochschule oder eine vergleichbare Prüfung abgelegt oder eine vergleichbare Qualifikation erworben haben (sachkundige Beisitzende). Die Prüfenden sind in ihrer Prüfungstätigkeit unabhängig. Die Prüfenden und die Beisitzenden unterliegen der Amtsverschwiegenheit.
- (2) Die/der Studierende kann einen oder mehrere Prüfer für die Betreuung der Masterarbeit vorschlagen. Auf diesen Vorschlag ist nach Möglichkeit Rücksicht zu nehmen. Der Prüfungsausschuss achtet darauf, dass die Prüfungsverpflichtung möglichst gleichmäßig auf die Prüfenden verteilt wird.
- (3) Das vorsitzende Mitglied des Prüfungsausschusses sorgt dafür, dass der/dem Studierenden die Namen der Prüfenden rechtzeitig bekannt gegeben werden. Die Bekanntgabe soll zugleich mit der Zulassung zur Prüfung, in der Regel mindestens zwei Wochen vor der Ausgabe der Masterarbeit, erfolgen. Die Bekanntmachung durch Aushang ist ausreichend.

§ 9

Anrechnung von Studienzeiten, Studien- und Prüfungsleistungen

- (1) Studienzeiten, Studien- und Prüfungsleistungen im gleichen Studiengang an anderen Hochschulen im Geltungsbereich des Grundgesetzes werden von Amts wegen angerechnet. Studien- und Prüfungsleistungen in anderen Studiengängen werden anerkannt, soweit die Gleichwertigkeit festgestellt ist. Gleichwertigkeit ist festzustellen, wenn
 - Studienzeiten, Studien- und Prüfungsleistungen in Inhalt, Umfang und in den Anforderungen denjenigen des Masterstudiengangs an der Fachhochschule Bielefeld im Wesentlichen entsprechen (§63, Abs. 2 Satz 2 HG) oder
 - keine wesentlichen Unterschiede zwischen den zu vergleichenden Zeiten (Art. V Ziff. 1 Lissabon-Konvention) bzw. Leistungen (Art. VI Ziff. 1 Lissabon-Konvention) bestehen.Dabei ist kein schematischer Vergleich, sondern eine Gesamtbetrachtung und Gesamtbewertung vorzunehmen.
- (2) Gleichwertige Studien- und Prüfungsleistungen an Hochschulen außerhalb des Geltungsbereichs des Grundgesetzes werden auf Antrag angerechnet. Für die Gleichwertigkeit sind die

von der Kultusministerkonferenz und Hochschulrektorenkonferenz gebilligten Äquivalenzvereinbarungen maßgebend. Soweit Äquivalenzvereinbarungen nicht vorliegen, entscheidet der Prüfungsausschuss über die Anrechnung. Bei Zweifeln in Fragen der Gleichwertigkeit werden die Prüfenden des Fachbereichs oder die Zentralstelle für ausländisches Bildungswesen beteiligt.

- (3) Sonstige Kenntnisse oder Qualifikationen werden auf Antrag auf der Grundlage vorgelegter Unterlagen angerechnet, sofern sie nicht bereits Voraussetzung für die Zulassung waren.
- (4) Fehlversuche in verwandten oder vergleichbaren Prüfungsleistungen sind anzurechnen. Alle Pflichtmodule sind in diesem Studiengang zu erbringen und können nicht als Fremdleistung in einem anderen Studiengang belegt und angerechnet werden.
- (5) Über die Anrechnung nach den Absätzen 1 bis 4 entscheidet der Prüfungsausschuss nach den Richtlinien des ECTS, immer nach Anhörung von den für die Fächer zuständigen Prüfenden.

§ 10

Wiederholung von Prüfungsleistungen

- (1) Eine nicht bestandene Modulprüfung kann zweimal wiederholt werden. Die Wiederholung soll zum nächsten Prüfungstermin nach Ableistung des erfolglosen Versuches stattfinden.
- (2) Die 2. Wiederholungsprüfung ist in jedem Fall von zwei Prüferinnen/Prüfern zu bewerten.
- (3) Die Masterarbeit kann einmal wiederholt werden.
- (4) Das Kolloquium zu einer Masterarbeit kann einmal wiederholt werden.
- (5) Eine mindestens als ausreichend bewertete Prüfungsleistung kann nicht wiederholt werden.

§ 11

Versäumnis, Rücktritt, Täuschung, Ordnungsverstoß

- (1) Eine Prüfungsleistung gilt als „nicht ausreichend“ (5,0) bewertet, wenn die/der Studierende zu einem Prüfungstermin ohne triftige Gründe nicht erscheint oder nach Beginn der Prüfung ohne triftige Gründe von der Prüfung zurücktritt oder die Prüfungsleistung nicht vor Ablauf der Prüfung erbringt. Satz 1 gilt entsprechend, wenn die Masterarbeit nicht fristgemäß abgeliefert wird. Wird die gestellte Prüfungsarbeit nicht bearbeitet, steht dies der Säumnis nach Satz 1 gleich. Belastende Entscheidungen sind den Betroffenen unverzüglich schriftlich mitzuteilen, zu begründen und mit einer Rechtsbehelfsbelehrung zu versehen.
- (2) Die für den Rücktritt oder das Versäumnis geltend gemachten Gründe müssen dem Prüfungsausschuss unverzüglich schriftlich angezeigt und glaubhaft gemacht werden. Bei Prüfungsunfähigkeit kann die Vorlage eines amtsärztlichen Attestes verlangt werden. Erkennt der Prüfungsausschuss die Gründe an, so kann die Zulassung zu der entsprechenden Prüfungsleistung erneut beantragt werden.
- (3) Versucht eine Studierende/ein Studierender, das Ergebnis einer Prüfungsleistung durch Täuschung oder Benutzung nicht zugelassener Hilfsmittel zu beeinflussen, gilt die betreffende Prüfungsleistung als „nicht ausreichend“ (5,0) bewertet. Wer als Studierende/Studierender den ordnungsgemäßen Ablauf der Prüfung stört, kann von der jeweiligen Aufsicht, in der Regel, nach Abmahnung von der Fortsetzung der Prüfungsleistung ausgeschlossen werden; in diesem Fall gilt die betreffende Prüfungsleistung als „nicht ausreichend“ (5,0) bewertet. Die Gründe für den Ausschluss sind aktenkundig zu machen. Wenn die/der Studierende davon ausgeschlossen wird, eine weitere Prüfungsleistung zu erbringen, kann sie/er verlangen, dass der Prüfungsausschuss diese Entscheidung überprüft. Dies gilt entsprechend auch bei den Feststellungen gemäß Satz 1.

II. Prüfungsabläufe

§ 12

Ziel, Umfang und Form der Modulprüfungen

- (1) Eine Modulprüfung ist eine studienbegleitende Prüfungsleistung. In den Modulprüfungen soll festgestellt werden, ob die Studierenden Inhalt und Methoden der Prüfungsmodule in den wesentlichen Zusammenhängen beherrschen und die erworbenen Kenntnisse, Fähigkeiten und Fertigkeiten selbständig anwenden können.
- (2) Die Prüfungsanforderungen sind an dem Inhalt der Lehrveranstaltungen und an den Qualifikationen zu orientieren, die für das betreffende Modul vorgesehen sind.
- (3) Eine Modulprüfung kann aus folgenden Leistungen bestehen:
 1. einer Klausur,

2. einer mündlichen Prüfung,
 3. einer schriftlichen Hausarbeit,
 4. einer Projektarbeit,
 5. einer Prüfung, in der in einer Verknüpfung zwischen praktischen und theoretischen Anteilen eine Fähigkeit aktuell entwickelt und verwirklicht wird ("Performanzprüfung").
- (4) Modulprüfungen können in Teilprüfungen zerlegt werden.
 - (5) In Modulen, in denen ein Teil des Lehrstoffes in Praktika vermittelt wird, ist für die Vergabe von Credits die Teilnahme durch eine bestandene Teilmodulprüfung (Testat) nachzuweisen. Ein Testat wird sowohl für die regelmäßige Teilnahme an den Praktika als auch für regelmäßig abzuliefernde Praktikumsaufgaben ausgestellt.
 - (6) Eine Teilmodulprüfung ist bestanden, wenn die Prüfungsleistung mindestens als ausreichend bewertet worden ist.
 - (7) Der Prüfungsausschuss legt in der Regel spätestens zwei Monate vor einem Prüfungstermin bzw. Prüfungszeitraum, frühestens aber zu Semesterbeginn, die Prüfungsform und den Umfang im Benehmen mit den Prüfenden für alle Kandidatinnen und Kandidaten der jeweiligen Modulprüfung einheitlich und verbindlich fest.

§ 13

Zulassung zu Modulprüfungen

- (1) An den jeweiligen Modulprüfungen darf nur teilnehmen, wer
 1. zum Zeitpunkt der Prüfung für den Studiengang als Ersthörender eingeschrieben oder gemäß § 52 Abs. 1 HG als Zweithörender zugelassen ist,
 2. die nach § 3 geforderten Voraussetzungen erfüllt,
 3. den Prüfungsanspruch in dem Studiengang oder in einem verwandten Studiengang nicht verloren hat.
- (2) Der Antrag auf Zulassung ist bis zu dem vom Prüfungsausschuss festgesetzten Termin schriftlich oder im Onlinesystem zu stellen. Der Antrag kann für mehrere Modulprüfungen zugleich gestellt werden, wenn diese Modulprüfungen innerhalb desselben Prüfungszeitraums oder die dafür vorgesehenen Prüfungstermine spätestens zu Beginn der Vorlesungszeit des folgenden Semesters stattfinden sollen.
- (3) Dem Antrag ist eine Erklärung darüber beizufügen, ob bei mündlichen Prüfungen einer Zulassung von Zuhörenden widersprochen wird. Ggf. muss diese bis zu einem vom Prüfungsamt festgesetzten Termin nachgereicht werden, sofern sie nicht bereits früher vorgelegt wurde.
- (4) Der Antrag auf Zulassung zu einer Modulprüfung kann schriftlich beim Prüfungsamt bis zum Ablauf des achten Tages vor dem festgesetzten Prüfungstermin ohne Anrechnung auf die Zahl der möglichen Prüfungsversuche zurückgenommen werden, so dass eine Frist von sieben Tagen besteht.
- (5) Über die Zulassung entscheidet das vorsitzende Mitglied des Prüfungsausschusses und im Zweifelsfall der Prüfungsausschuss.
- (6) Die Zulassung ist zu versagen, wenn
 1. die in den Absätzen 1 bis 2 genannten Voraussetzungen nicht erfüllt sind oder
 2. die Unterlagen unvollständig sind und nicht bis zu dem vom Prüfungsamt festgesetzten Termin ergänzt werden oder
 3. eine entsprechende Modulprüfung in einem Masterstudiengang oder in einem verwandten Studiengang endgültig nicht bestanden wurde. Dies gilt entsprechend für eine Masterprüfung im Geltungsbereich des Grundgesetzes.
- (7) Im Übrigen darf die Zulassung nur versagt werden, wenn die/der Studierende im Geltungsbereich des Grundgesetzes ihren/seinen Prüfungsanspruch im gleichen Studiengang durch Versäumen einer Wiederholungsfrist verloren hat.
- (8) Die oder der Studierende ist verpflichtet, sich über die Anmelde- und Prüfungszeiträume sowie die Zulassung bzw. Nicht-Zulassung zu informieren und die Aushänge zu beachten.

§ 14

Durchführung von Modulprüfungen

- (1) Für die Prüfungen der Pflichtmodule sind in jedem Studienjahr mindestens zwei Prüfungstermine anzusetzen; für die Prüfungen der Wahlpflichtmodule wird ein erster Prüfungszeitraum festgesetzt, der in dem Semester liegt, in welchem das Modul angeboten wurde, sowie zwei weitere innerhalb der zwei nächstfolgenden Semester. Die Modulprüfungen sollen innerhalb der Prüfungszeiträume stattfinden, die vom Prüfungsausschuss festgesetzt und bei Semesterbeginn oder zum Ende des vorhergehenden Semesters bekannt gegeben werden.

- (2) Der Prüfungstermin wird den Studierenden rechtzeitig, in der Regel mindestens zwei Wochen vor der betreffenden Prüfung, bekannt gegeben. Die Bekanntmachung durch Aushang ist ausreichend.
- (3) Die/der Studierende hat sich auf Verlangen der aufsichtführenden Person mit einem amtlichen Ausweis auszuweisen.
- (4) Macht die/der Studierende durch ein ärztliches Zeugnis oder auf andere Weise glaubhaft, dass sie/er wegen ständiger körperlicher Behinderung und/oder chronischer Erkrankung nicht in der Lage ist, die Prüfung ganz oder teilweise in der vorgesehenen Form abzulegen, kann gestattet werden, gleichwertige Prüfungsleistungen in einer anderen Form zu erbringen. Es ist dafür zu sorgen, dass durch die Gestaltung der Prüfungsbedingungen eine Benachteiligung für behinderte und/oder chronisch kranke Menschen nach Möglichkeit ausgeglichen wird. Im Zweifel können weitere Nachweise angefordert werden.
- (5) Das Prüfungsergebnis wird dem Prüfungsamt durch den Prüfenden entsprechend der für die jeweilige Prüfungsform festgelegten Art und Weise innerhalb des festgelegten Zeitrahmens mitgeteilt.
- (6) Den Studierenden ist die Bewertung von Prüfungen und der Masterarbeit nach spätestens sechs Wochen mitzuteilen. Die Bekanntmachung erfolgt im Onlinesystem und durch Aushang.

§ 15

Klausurarbeiten und sonstige schriftliche Arbeiten

- (1) In den Klausurarbeiten sollen Studierende nachweisen, dass sie in begrenzter Zeit theoretische Grundlagen darstellen und mit beschränkten Hilfsmitteln Probleme aus Gebieten des jeweiligen Moduls mit geläufigen Methoden der Fachrichtung erkennen und stringent zu einer Lösung führen können.
- (2) Eine Klausurarbeit findet unter Aufsicht statt. Über die Zulassung von Hilfsmitteln entscheiden die Prüfenden. Die Dauer einer Klausurarbeit soll 60 Minuten nicht unter- und 180 Minuten nicht überschreiten.
- (3) Die Prüfungsaufgabe einer Klausurarbeit wird in der Regel von nur einer prüfenden Person gestellt. In fachlich begründeten Fällen, insbesondere wenn in einer Modulprüfung mehrere Fachgebiete zusammenfassend geprüft werden, kann die Prüfungsaufgabe auch von mehreren Prüfenden gestellt werden. In diesem Fall legen die Prüfenden die Gewichtung der Anteile an der Prüfungsaufgabe vorher gemeinsam fest.
- (4) Die Bewertung von Klausurarbeiten durch eine Prüferin oder einen Prüfer ist ausreichend. In den Fällen des Absatzes 3 Satz 2 bewerten die Prüfenden in der Regel nur den eigenen Aufgabenteil; Satz 1 bleibt unberührt.

§ 16

Mündliche Prüfungen

- (1) Durch mündliche Prüfungsleistungen sollen Studierende nachweisen, dass sie die Zusammenhänge des Prüfungsgebietes erkennen und spezielle Fragestellungen in diese Zusammenhänge einzuordnen vermögen. Ferner soll festgestellt werden, ob die Studierenden über ein breites Grundlagenwissen verfügen. Die Dauer der Prüfung beträgt je Studierenden mindestens 15 und höchstens 45 Minuten. Die prüfende Person kann der/dem Studierenden eine angemessene Vorbereitungszeit, die Bestandteil der Prüfung ist, aber nicht auf deren Dauer angerechnet wird, einräumen.
- (2) Mündliche Prüfungen sind von mindestens zwei Prüfenden (Kollegialprüfung) oder von einem Prüfenden in Gegenwart einer/eines sachkundigen Beisitzenden als Gruppenprüfung oder als Einzelprüfung abzunehmen. Hierbei wird jede/r Studierende in einer Modulprüfung im Regelfall nur von einer Person geprüft. Vor der Festsetzung der Note hat die prüfende Person die anderen an der Prüfung mitwirkenden Prüfer/in beziehungsweise die/den sachkundigen Beisitzenden zu hören.
- (3) Die sachkundigen Beisitzenden haben während der Prüfung kein Fragerecht.
- (4) Die wesentlichen Gegenstände und Ergebnisse der Prüfung, insbesondere die für die Benotung maßgeblichen Tatsachen, sind in einem Protokoll festzuhalten. Das Ergebnis der Prüfung ist der/dem Studierenden im Anschluss an die mündliche Prüfung bekannt zu geben. Bei der Bekanntgabe des Ergebnisses sind die Bestimmungen des Datenschutzes zu beachten.
- (5) Studierende, die sich in einem späteren Prüfungszeitraum der gleichen Prüfung unterziehen wollen, werden nach Maßgabe der räumlichen Verhältnisse als Zuhörende zugelassen, sofern nicht bei der Meldung zur Prüfung widersprochen wird. Die Zulassung erstreckt sich nicht auf die Beratung und Bekanntgabe des Prüfungsergebnisses.

§ 17

Hausarbeiten

- (1) Hausarbeiten sind Ausarbeitungen, die in der Regel 20 Seiten nicht überschreiten und die im Rahmen einer Lehrveranstaltung oder in Verbindung mit einer Projektarbeit begleitend zu dieser erstellt werden. Sie können je nach Maßgabe des Lehrenden durch einen Fachvortrag in Form einer mündlichen Prüfung (geregelt wie in § 16 Abs. 2-5) von in der Regel 15 bis 45 Minuten Dauer ergänzt werden.
- (2) In Hausarbeiten sollen die Studierenden in begrenzter Zeit nachweisen, dass sie die Zusammenhänge des Moduls im jeweiligen Fachgebiet erkennen, spezielle Fragestellungen in diese Zusammenhänge einzuordnen vermögen und stringent fachspezifische Probleme lösen können.
- (3) Über Art, Umfang, zeitlichen Rahmen und Ausführung der Hausarbeit entscheidet die/der Lehrende im Rahmen der Maßgabe des Absatzes 1.
- (4) Die Hausarbeit ist innerhalb einer von der/dem Lehrenden festzulegenden Frist beim zuständigen Prüfungsamt abzuliefern. Die Frist ist durch Aushang bekannt zu machen. Bei der Abgabe der Hausarbeit hat die/der Studierende zu versichern, dass sie/er ihre/seine Arbeit – bei einer Gruppenarbeit ihren/seinen gekennzeichneten Anteil der Arbeit – selbständig angefertigt und keine anderen als die angegebenen und bei Zitaten kenntlich gemachten Hilfsmittel benutzt hat. Der Abgabezeitpunkt der schriftlichen Hausarbeit ist aktenkundig zu machen. Bei Zustellung der Arbeit durch die Post ist der Zeitpunkt der Einlieferung bei der Post maßgebend. Wird die Hausarbeit nicht fristgemäß abgeliefert, gilt sie als mit „nicht ausreichend“ (5,0) bewertet.
- (5) Die Bewertung der Hausarbeit durch eine/n Prüfenden ist ausreichend.

§ 18

Projektarbeiten

- (1) Die Projektarbeit besteht aus einer schriftlichen Ausarbeitung, einer Übergabe einer ausgearbeiteten Software (Lösung) und ggfs. – nach Angabe der/des Lehrenden - einer Präsentation.
- (2) Ein Projekt ist eine Aufgabe, die von der/dem Lehrenden in Zusammenarbeit mit den Studierenden nach Möglichkeit interdisziplinär geplant und ausgewählt wird. Die Durchführung erfolgt möglichst selbständig unter Beratung durch eine/n oder mehrere Lehrende, die auch interdisziplinär zusammengestellt sein können. In einem Projekt werden konkrete Problemstellungen ganzheitlich und unter praxisnahen Bedingungen bearbeitet. Bei Gruppenarbeiten werden die inhaltliche und gleichmäßige Verteilung der Arbeitsinhalte an die Studierenden durch die Lehrenden vorgenommen.
- (3) Die Prüfungsleistungen der/des einzelnen Studierenden werden nach Abschluss des jeweiligen Semesters von der/dem zuständigen Lehrenden nach einer Auswahl aus den Kriterien:
 - Dokumentation
 - Qualität der ausgearbeiteten Software
 - Ggf. Präsentation durch die einzelne Studierende/den einzelnen Studierenden
 - Ggf. Beitrag zum Teamergebnis bei einer Gruppenarbeit
 - Ggf. Teamfähigkeitbewertet. Die Ergebnisse werden in einer Liste erfasst.
- (4) Die Prüfung der Projektarbeit kann durch eine Präsentation von 30 bis 45 Minuten abgelegt werden. Bei Gruppenarbeiten sind von allen am jeweiligen Projekt beteiligten Studierenden die Einzelbeiträge und Ergebnisse vorzutragen. Die Präsentation findet in Gegenwart der/des Lehrenden, die/der die Projektarbeit begleitet hat, statt. § 16 Abs. 2 bis 5 sind auf die Präsentation entsprechend anzuwenden.

§ 19

Performanzprüfungen

- (1) In fachlich geeigneten Fällen kann eine Modulprüfung durch eine Performanzprüfung abgelegt werden. Im Rahmen einer Performanzprüfung werden von der/dem Studierenden erworbenes Wissen und die angelegten Kompetenzen exemplarisch auf die zukünftige Berufspraxis angewendet. Es soll festgestellt werden, ob die Studierenden ihr theoretisches Wissen praktisch anwenden können.
- (2) Eine Performanzprüfung ist dadurch gekennzeichnet, dass sie sich aus zwei Anteilen (theoretisch und praktisch) zusammensetzt. Eine Teilleistung ist bestanden, wenn sie mindestens mit ausreichend bewertet worden ist. Der theoretische Anteil besteht aus einer schriftlichen oder mündlichen Prüfungsleistung entsprechend der §§ 15 und 16. Der praktische Anteil besteht aus praxisnahen Aufgaben, welche im Laufe des Semesters absolviert und bewertet werden.

Einzelne bestandene Anteile können auf die Folgesemester übertragen werden. Jede Teilleistung wird in der Regel von nur einer prüfenden Person durchgeführt.

- (3) Die wesentlichen Gegenstände und Ergebnisse der jeweiligen Teilleistung, insbesondere die für die Benotung maßgeblichen Tatsachen, sind in einem Protokoll festzuhalten. Das Ergebnis der Prüfung ist der/dem Studierenden im Anschluss an die Teilleistung bekannt zu geben. Bei der Bekanntgabe des Ergebnisses sind die Bestimmungen des Datenschutzes zu beachten.
- (4) Die Gesamtnote ergibt sich als arithmetisches Mittel aus den Bewertungen der beiden Einzelleistungen.

§ 20

Abzuleistende Modulprüfungen, Credits

Der Studienverlaufsplan (Anlage 1) legt fest, welche Pflicht- und welche Wahlpflichtmodule mit einer Prüfung abzuschließen sind. Er ordnet auch die entsprechenden Credits zu.

§ 21

Bewertung von Prüfungsleistungen

- (1) Prüfungsleistungen sind durch Noten differenziert zu beurteilen. Die Noten für die einzelnen Prüfungsleistungen werden von den jeweiligen Prüfenden festgesetzt.
- (2) Sind mehrere Prüfende an einer Prüfung beteiligt, so bewerten sie die gesamte Prüfungsleistung gemeinsam, sofern nicht nachfolgend etwas anderes bestimmt ist. Bei nicht übereinstimmender Beurteilung ergibt sich die Note aus dem arithmetischen Mittel der Einzelbewertungen.
- (3) Für die Bewertung der Prüfungsleistungen sind folgende Noten zu verwenden:
 - 1 = sehr gut = eine hervorragende Leistung;
 - 2 = gut = eine Leistung, die erheblich über den durchschnittlichen Anforderungen liegt;
 - 3 = befriedigend = eine Leistung, die durchschnittlichen Anforderungen entspricht;
 - 4 = ausreichend = eine Leistung, die trotz ihrer Mängel noch den Anforderungen genügt;
 - 5 = nicht ausreichend = eine Leistung, die wegen erheblicher Mängel den Anforderungen nicht mehr genügt.

Zur weiteren Differenzierung der Bewertung können um 0,3 verminderte oder erhöhte Notenziffern gebildet werden; die Noten 0,7, 4,3, 4,7 und 5,3 sind ausgeschlossen.

- (4) Besteht eine Prüfung aus mehreren Prüfungsleistungen, errechnet sich die Note aus dem nach Credits gewichteten Durchschnitt (gewichtetes arithmetisches Mittel) der Noten der einzelnen Prüfungsleistungen. Bei einer Mittelung von Noten ergibt sich die Gesamtnote wie folgt:
 - bei einem Durchschnitt bis 1,5 = die Note „sehr gut“
 - bei einem Durchschnitt von 1,6 bis 2,5 = die Note „gut“
 - bei einem Durchschnitt von 2,6 bis 3,5 = die Note „befriedigend“
 - bei einem Durchschnitt von 3,6 bis 4,0 = die Note „ausreichend“
 - bei einem Durchschnitt ab 4,1 = die Note „nicht ausreichend“.

Hierbei werden Zwischenwerte nur mit der ersten Dezimalstelle berücksichtigt; alle weiteren Stellen hinter dem Komma werden ohne Rundung gestrichen.

- (5) Für die Umrechnung von Noten in ECTS-Grades bei der Abschlussnote wird, sobald eine ausreichende Zahl von Absolventinnen und Absolventen vorhanden ist, die folgende Tabelle zugrunde gelegt:

A	= die besten 10%
B	= die nächsten 25%
C	= die nächsten 30%
D	= die nächsten 25%
E	= die nächsten 10%
FX/F	= nicht bestanden, es sind (erhebliche) Verbesserungen erforderlich.
- (6) Für jede bestandene Modulprüfung werden Credits nach Maßgabe der Anlage 1 vergeben.

III. Masterarbeit (Thesis) und Kolloquium

§ 22

Masterarbeit (Thesis)

- (1) Die Masterarbeit hat zu zeigen, dass die/der Studierende befähigt ist, innerhalb einer vorgegebenen Frist eine praxisorientierte Forschungs- und Entwicklungsaufgabe aus dem Fachgebiet, sowohl in ihren fachlichen Einzelheiten als auch in den fachübergreifenden Zusammenhängen nach wissenschaftlichen Methoden selbständig zu bearbeiten. Die Masterarbeit ist eine schriftliche Hausarbeit. Sie besteht in der Regel aus der Analyse, Planung, Implementierung und Evaluation eines Softwareentwicklungsprojektes in Einrichtungen, die mit den Zielen und Inhalten des Studienganges in einem fachlichen Zusammenhang stehen. Die Masterarbeit ist eine eigenständige wissenschaftliche Arbeit aus dem Themenumfeld der Informatik. Der Umfang der Masterarbeit soll 60 Textseiten nicht unterschreiten und 80 Textseiten nicht überschreiten.
- (2) Die Masterarbeit kann von jeder prüfenden Person, welche die Voraussetzungen gemäß § 8 erfüllt, ausgegeben und betreut werden. Auf Antrag der/des Studierenden kann der Prüfungsausschuss auch eine Honorarprofessorin oder einen Honorarprofessor oder mit entsprechenden Aufgaben betraute Lehrbeauftragte gem. § 8 Abs. 1 mit der Betreuung bestellen, wenn feststeht, dass das vorgesehene Thema der Masterarbeit nicht durch eine fachlich zuständige Professorin oder einen fachlich zuständigen Professor betreut werden kann. Die Masterarbeit darf mit Zustimmung des Prüfungsausschusses in einer Einrichtung außerhalb der Hochschule durchgeführt werden, wenn sie dort ausreichend betreut werden kann. Den Studierenden ist die Gelegenheit zu geben, Vorschläge für den Themenbereich der Masterarbeit zu machen.
- (3) Die Masterarbeit kann auch in Form einer Gruppenarbeit zugelassen werden, wenn der als Prüfungsleistung zu bewertende Beitrag der/des einzelnen Studierenden deutlich unterscheidbar und bewertbar ist und die Anforderungen nach Abs. 1 erfüllt sind. Hierzu ist eine eindeutige Abgrenzung durch die Angabe von Abschnitten, Seitenzahlen oder anderen objektiven Kriterien erforderlich.

§ 23

Zulassung zur Masterarbeit

- (1) Zur Masterarbeit wird zugelassen, wer mindestens 50 Credits erworben hat.
- (2) Der Antrag auf Zulassung ist schriftlich an den Prüfungsausschuss zu richten. Dem Antrag sind folgende Unterlagen beizufügen, sofern sie nicht bereits früher vorgelegt wurden:
 1. die Nachweise über die in Absatz 1 genannten Zulassungsvoraussetzungen,
 2. eine Erklärung über bisherige Versuche zur Bearbeitung einer Masterarbeit. Dem Antrag soll eine Erklärung darüber beigefügt werden, welche prüfende Person zur Ausgabe und Betreuung der Masterarbeit bereit ist.
- (3) Der Antrag auf Zulassung kann schriftlich bis zur Bekanntgabe der Entscheidung über den Antrag ohne Anrechnung auf die Zahl der möglichen Prüfungsversuche zurückgenommen werden.
- (4) Über die Zulassung entscheidet das vorsitzende Mitglied des Prüfungsausschusses und im Zweifelsfall der Prüfungsausschuss. Die Zulassung ist zu versagen, wenn
 3. die in Absatz 1 genannten Voraussetzungen nicht erfüllt oder
 4. die Unterlagen unvollständig sind oder
 5. im Geltungsbereich des Grundgesetzes eine entsprechende Masterarbeit ohne Wiederholungsmöglichkeit als "nicht ausreichend" bewertet worden ist oder eine in der Anlage 1 genannte Prüfung endgültig nicht bestanden wurde.

Im Übrigen darf die Zulassung nur versagt werden, wenn die/der Studierende im Geltungsbereich des Grundgesetzes ihren/seinen Prüfungsanspruch im gleichen Studiengang durch Versäumen einer Wiederholungsfrist verloren hat.

§ 24

Ausgabe und Bearbeitung der Masterarbeit

- (1) Die Ausgabe des Themas der Masterarbeit und die Festlegung der Bearbeitungsdauer erfolgen durch den Prüfungsausschuss. Als Zeitpunkt der Ausgabe gilt der Tag, an dem das Prüfungsamt das von der betreuenden Person gestellte Thema der Masterarbeit der/dem Studierenden bekannt gibt; der Zeitpunkt ist aktenkundig zu machen.

- (2) Die Bearbeitungszeit (Zeitraum von der Ausgabe bis zur Abgabe der Masterarbeit) beträgt höchstens fünf Monate. Das Thema und die Aufgabenstellung müssen so beschaffen sein, dass die Masterarbeit innerhalb der vorgesehenen Frist abgeschlossen werden kann. Wird die Masterarbeit nicht fristgerecht abgegeben, gilt sie als mit „nicht ausreichend“ (5,0) bewertet. Im Ausnahmefall kann das Prüfungsamt auf einen vor Ablauf der Frist gestellten Antrag die Bearbeitungszeit um bis zu vier Wochen verlängern. Die Person, welche die Masterarbeit betreut, soll zu dem Antrag gehört werden.
- (3) Das Thema der Masterarbeit kann nur einmal und nur innerhalb der ersten zwei Wochen der Bearbeitungszeit ohne Angabe von Gründen zurückgegeben werden. Im Fall der Wiederholung gemäß § 10 ist die Rückgabe nur zulässig, wenn bei der Anfertigung der ersten Masterarbeit von dieser Möglichkeit kein Gebrauch gemacht worden ist.
- (4) Über die Zulassung entscheidet das vorsitzende Mitglied des Prüfungsausschusses und im Zweifelsfall der Prüfungsausschuss.

§ 25

Abgabe und Bewertung der Masterarbeit

- (1) Die Masterarbeit ist fristgemäß beim Prüfungsamt abzuliefern. Der Zeitpunkt der Abgabe ist aktenkundig zu machen; bei Zustellung der Arbeit durch ein Beförderungsunternehmen ist der Zeitpunkt der Einlieferung bei dem Beförderungsunternehmen maßgebend. Bei der Abgabe der Masterarbeit ist schriftlich zu versichern, dass die Arbeit - bei einer Gruppenarbeit der entsprechend gekennzeichnete Anteil der Arbeit - selbständig angefertigt wurde und keine anderen als die angegebenen und bei Zitaten kenntlich gemachten Quellen und Hilfsmittel benutzt worden sind.
- (2) Die Masterarbeit ist von zwei Personen zu bewerten, von denen eine die Masterarbeit betreut haben soll. Die zweite prüfende Person wird vom Prüfungsausschuss bestimmt. Mindestens eine der prüfenden Personen muss ein Vertreter der Professorenschaft sein, die im Masterstudiengang lehrt. Bei nicht übereinstimmender Bewertung durch die Prüfenden wird die Note der Masterarbeit aus dem arithmetischen Mittel der Einzelbewertungen gebildet, wenn die Differenz der beiden Noten weniger als 2,0 beträgt. Beträgt die Differenz 2,0 oder mehr, wird vom Prüfungsausschuss eine dritte prüfende Person bestimmt. In diesem Fall ergibt sich die Note der Masterarbeit aus dem arithmetischen Mittel der drei Einzelbewertungen. Die Masterarbeit kann jedoch nur dann als "ausreichend" (4,0) oder besser bewertet werden, wenn mindestens zwei der Noten "ausreichend" (4,0) oder besser sind. Alle Bewertungen sind schriftlich zu begründen.
- (3) Für eine mindestens ausreichend zu bewertende Masterarbeit werden 24 Credits vergeben.

§ 26

Kolloquium

- (1) Das Kolloquium ergänzt die Masterarbeit und ist selbständig zu bewerten. Es dient der Feststellung, ob die/der Studierende befähigt ist, die Ergebnisse der Masterarbeit, ihre fachlichen Grundlagen, ihre fachübergreifenden Zusammenhänge und ihre außerfachlichen Bezüge mündlich darzustellen und selbständig zu begründen und ihre Bedeutung für die Praxis einzuschätzen. Dabei soll auch die Bearbeitung des Themas der Masterarbeit mit dem Prüfling erörtert werden.
- (2) Zu Beginn des Kolloquiums soll die Masterarbeit in einem mündlichen Vortrag präsentiert werden.
- (3) Die Zulassung zum Kolloquium erfolgt nur, wenn
 1. alle bis auf zwei studienbegleitenden Prüfungen bestanden sind,
 2. die Masterarbeit mindestens mit der Note 4,0 bewertet worden ist.
 Der Antrag auf Zulassung ist an den Prüfungsausschuss zu richten. Dem Antrag sind die Nachweise über die in Satz 1 genannten Zulassungsvoraussetzungen beizufügen, sofern sie dem Prüfungsausschuss nicht bereits vorliegen; ferner ist eine Erklärung über bisherige Versuche zur Ablegung entsprechender Prüfungen sowie darüber, ob einer Zulassung von Zuhörenden widersprochen wird, beizufügen. Die Zulassung zum Kolloquium kann auch bereits bei der Meldung zur Masterarbeit beantragt werden; in diesem Fall erfolgt die Zulassung zum Kolloquium, sobald alle erforderlichen Nachweise und Unterlagen dem Prüfungsausschuss vorliegen. Für die Zulassung zum Kolloquium und ihre Versagung gilt im übrigen § 25 Abs. 4 entsprechend.
- (4) Das Kolloquium wird als mündliche Prüfung durchgeführt.

- (5) Das Kolloquium wird von den Prüfenden der Masterarbeit gemeinsam abgenommen und bewertet. Im Fall des § 27 Abs. 2 wird das Kolloquium von den Prüfenden abgenommen, aus deren Einzelbewertung die Note der Masterarbeit gebildet worden ist.
- (6) Das Kolloquium dauert zusammen mit dem Vortrag mindestens 45 Minuten und höchstens 60 Minuten. Für die Durchführung des Kolloquiums finden im Übrigen die für die mündlichen Prüfungen geltenden Vorschriften entsprechend Anwendung.
- (7) Abweichend von den mündlichen Prüfungen ist das Kolloquium grundsätzlich eine fachhochschuloffene Veranstaltung, dies erstreckt sich nicht auf die Bekanntgabe der Note.
- (8) Liegen Gründe für eine vertrauliche Behandlung der Darstellung der Ergebnisse der Masterarbeit im Kolloquium vor, entscheidet der Prüfungsausschuss auf Antrag eines der Betreuer der Masterarbeit oder der/des Studierenden über den Ausschluss der Öffentlichkeit.
- (9) Personen, die in einem inhaltlichen Zusammenhang mit der Masterarbeit stehen (z.B. als externer Mitbetreuer), können vom Vorsitzenden des Prüfungsausschusses zum Kolloquium auf Antrag zugelassen werden, sofern der Absatz 8 dem nicht widerspricht.
- (10) Für ein mindestens ausreichend zu bewertendes Kolloquium werden 6 Credits vergeben.

IV. Ergebnis der Masterprüfung, Zusatzmodule

§ 27

Ergebnis der Masterprüfung

- (1) Die Masterprüfung ist bestanden, wenn 90 Credits erreicht wurden.
- (2) Die Masterprüfung ist nicht bestanden, wenn
 1. die Gesamtnote nicht mindestens „ausreichend“ (4,0) ist oder
 2. die Masterarbeit im zweiten Versuch nicht bestanden ist oder als nicht bestanden gilt.
- (3) Wird die Masterprüfung nicht bestanden, ist ein Bescheid zu erteilen, der mit einer Belehrung über den Rechtsbehelf zu versehen ist.
- (4) Studierende, welche die Hochschule ohne Masterabschluss verlassen, erhalten auf Antrag ein Zeugnis über die erbrachten Studien- und Prüfungsleistungen gemäß § 66 Abs. 4 HG.

§ 28

Zeugnis, Gesamtnote, Masterurkunde, Diploma Supplement

- (1) Über die bestandene Masterprüfung wird unverzüglich, möglichst innerhalb von vier Wochen nach Bekanntgabe des Ergebnisses, ein Zeugnis ausgestellt. Das Zeugnis enthält die Noten und Credit Points der Modulprüfungen, das Thema und die Note der Masterarbeit sowie die Gesamtnote der Masterprüfung.
- (2) Zur Ermittlung der Gesamtnote für das Masterstudium werden die Noten für die einzelnen benoteten Prüfungsleistungen mit den jeweiligen ausgewiesenen Credits multipliziert. Die Summe der gewichteten Noten wird anschließend durch die Gesamtzahl der einbezogenen Credits dividiert.
- (3) Das Zeugnis ist von dem vorsitzenden Mitglied des Prüfungsausschusses zu unterzeichnen und trägt das Datum des Tages, an dem die letzte Prüfungsleistung erbracht worden ist.
- (4) Zusätzlich zu dem Zeugnis erhält der Kandidat die Masterurkunde mit dem Datum des Zeugnisses. Darin wird die Verleihung des Mastergrades gemäß § 2 Abs. 4 beurkundet. Die Masterurkunde wird von der Präsidentin bzw. dem Präsidenten der Fachhochschule Bielefeld unterzeichnet und mit deren Siegel versehen.
- (5) Zusätzlich erhält der Kandidat ein in englischer Sprache ausgestelltes Diploma Supplement mit dem Datum des Zeugnisses. Das Diploma Supplement wird vom vorsitzenden Mitglied des Prüfungsausschusses unterzeichnet.
- (6) Urkunden über Hochschulgrade können mehrsprachig ausgestellt werden (§ 66 Abs. 3 HG).

§ 29

Zusatzmodule

Die Studierenden können sich in weiteren als den vorgeschriebenen Modulen einer Prüfung unterziehen. Das Ergebnis dieser Modulprüfungen wird eine Anlage des Zeugnisses aufgenommen, jedoch bei der Festsetzung der Gesamtnote nicht berücksichtigt.

V. Schlussbestimmungen

§ 30

Einsicht in die Prüfungsakte

- (1) Nach Abschluss des Prüfungsverfahrens wird den Studierenden auf Antrag Einsicht in ihre schriftlichen Prüfungsarbeiten, die darauf bezogenen Gutachten der Prüfer/innen und in die Prüfungsprotokolle gewährt.
- (2) Die Einsichtnahme ist binnen eines Jahres nach Aushändigung des Prüfungszeugnisses oder des Bescheides über die nicht bestandene Masterprüfung zu beantragen. § 32 des Verwaltungsverfahrensgesetzes für das Land Nordrhein-Westfalen über die Wiedereinsetzung in den vorigen Stand gilt entsprechend. Der Antrag ist bei dem vorsitzenden Mitglied des Prüfungsausschusses zu stellen. Dieser bestimmt Ort und Zeit der Einsichtnahme.
- (3) Die Einsichtnahme in die Prüfungsunterlagen, die sich auf eine Modulprüfung oder eine ergänzende Studienleistung beziehen, wird auf Antrag bereits nach Ablegung der jeweiligen Prüfung gestattet. Der Antrag ist binnen eines Monats nach Bekanntgabe des Prüfungsergebnisses zu stellen. Im Übrigen gilt Abs. 2 entsprechend.

§ 31

Ungültigkeit von Prüfungen

- (1) Hat eine Studierende/ein Studierender bei einer Prüfung getäuscht und wird diese Tatsache erst nach Aushändigung des Zeugnisses und der Urkunde bekannt, so kann der Prüfungsausschuss nachträglich die betroffenen Noten entsprechend berichtigen und die Prüfung ganz oder teilweise für nicht bestanden erklären.
- (2) Waren die Voraussetzungen für die Zulassung zu einer Prüfung nicht erfüllt, ohne dass die/der Studierende hierüber täuschen wollte und wird diese Tatsache erst nach Aushändigung des Zeugnisses und der Urkunde bekannt, so wird dieser Mangel durch das Bestehen der Prüfung geheilt. Wurde die Zulassung vorsätzlich zu Unrecht erwirkt, so entscheidet der Prüfungsausschuss unter Beachtung des Verwaltungsverfahrensgesetzes des Landes Nordrhein-Westfalen über die Rechtsfolgen.
- (3) Den Betroffenen ist vor einer Entscheidung Gelegenheit zur Äußerung zu geben.
- (4) Das unrichtige Prüfungszeugnis und die Urkunde sind einzuziehen und gegebenenfalls neu zu erteilen. Eine Entscheidung nach Absatz 1 und Absatz 2 Satz 2 ist nach einer Frist von fünf Jahren ab dem Datum des Prüfungszeugnisses und der Urkunde ausgeschlossen.

§ 32

In-Kraft-Treten, Veröffentlichung

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

Bielefeld, 17.06.2013

Die Präsidentin
der Fachhochschule Bielefeld

Prof. Dr. B. Rennen-Allhoff

Anlage 1: Studienverlaufsplan
Masterstudiengang Informatik

1. bzw. 2. Semester (Sommer)	cps	SWS	V	SU	Ü	P
Spezielle Gebiete zum Softwareengineering	10	6	-	4	-	2
Business Engineering und IT-Projektmanagement	5	4	-	4	-	-
Wahlfach aus Masterliste 1 „Fortgeschrittene Methoden der Informatik“	5	4	-	2	-	2
Wahlfach aus Masterliste 2 „Wissenschaftliche Anwendungen der Informatik“	10	6	-	2	-	4
Summen	30	20	-	12	-	8
1. bzw. 2. Semester (Winter)	cps	SWS	V	SU	Ü	P
Komplexitätstheorie	5	4	-	2	2	-
Formale Modelle der Informatik	5	4	-	2	-	2
Zuverlässige und sichere Softwaresysteme	5	4	-	2	2	-
Wahlfach aus Masterliste 1 „Fortgeschrittene Methoden der Informatik“	5	4	-	2	-	2
Wahlfach aus Masterliste 2 „Wissenschaftliche Anwendungen der Informatik“	10	6	-	2	-	4
Summen	30	22	-	10	4	8
3. Semester	cps	SWS	V	SU	Ü	P
Masterarbeit	24	-	-	-	-	-
Kolloquium	6	-	-	-	-	-
Summen	30	-	-	-	-	-
Gesamtsummen	90	42	-	22	4	16

Masterliste 1

„Fortgeschrittene Methoden der Informatik“

(2 V, 2 P)

- Architekturen, Modelle und Sprachen moderner Informationssysteme
- Grafische Datenverarbeitung
- Methoden der praktischen Optimierung
- Compilerbau
- Moderne Datenbanksysteme
- Effiziente Algorithmen
- Innovationsmanagement und Technologietransfer
- Mobile Anwendungen
- Operations Research
- Theoretische Kryptographie
- Fortgeschrittene Methoden der Künstlichen Intelligenz
- Web-Application-Security
- Funktionale Programmierung
- Public Key Infrastructure

Masterliste 2

„Wissenschaftliche Anwendungen der Informatik“

(2 V, 4 P)

- Computer Vision
- Visualisierung
- Hardwarenahe Programmierung
- Design und Modellierung von Softwaresystemen
- Intelligente Systeme
- Systemsicherheit
- Data Science
- Distributed Organic Computing Systems
- Verteilte und eingebettete Systeme
- Umweltinformatik
- Musikinformatik
- Domainspezifische Sprachen

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

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

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

Anlage 2: Modulhandbuch

Masterstudiengang Informatik

Inhaltsverzeichnis

Spezielle Gebiete zum Software Engineering	19
Business Engineering und IT-Projektmanagement	21
Masterliste 1: Architekturen, Modelle und Sprachen moderner Informationssysteme	22
Masterliste 1: Grafische Datenverarbeitung	24
Masterliste 1: Methoden der praktischen Optimierung	25
Masterliste 1: Compilerbau	26
Masterliste 1: Moderne Datenbanksysteme	27
Masterliste 1: Effiziente Algorithmen	28
Masterliste 1: Innovationsmanagement und Technologietransfer	29
Masterliste 1: Mobile Anwendungen	30
Masterliste 1: Operations Research	31
Masterliste 1: Theoretische Kryptographie	32
Masterliste 1: Fortgeschrittene Methoden der Künstlichen Intelligenz	33
Masterliste 1: Web-Application-Security	35
Masterliste 1: Funktionale Programmierung	36
Masterliste 1: Public Key Infrastructure	37
Masterliste 2: Computer Vision	39
Masterliste 2: Visualisierung	40
Masterliste 2: Hardwarenahe Programmierung	41
Masterliste 2: Design und Modellierung von Softwaresystemen	42
Masterliste 2: Intelligente Systeme	44
Masterliste 2: Systemsicherheit	46
Masterliste 2: Data Science	48
Masterliste 2: Distributed Organic Computing Systems	50
Masterliste 2: Verteilte und eingebettete Systeme	52
Masterliste 2: Umweltinformatik	53
Masterliste 2: Musikinformatik	55
Masterliste 2: Domainspezifische Sprachen	57
Komplexitätstheorie	59
Formale Modelle der Informatik	60
Zuverlässige und sichere Softwaresysteme	62
Masterarbeit	63
Kolloquium	64

Spezielle Gebiete zum Software Engineering					
Kenn-Nr. 1.1	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS Praktikum 2 SWS	Kontaktzeit 6 SWS /90 h	Selbststudium 210 h	geplante Gruppengröße Seminaristischer Unterricht 35 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Im Bereich technologischer Kompetenzen sollen tiefgehender Fachkenntnisse in mehreren ausgewählten Spezialgebieten des Software Engineerings erworben werden und ein vertieftes wissenschaftliches Verständnis für anwendbare Techniken und Methoden sowie deren Grenzen erworben werden.</p> <p>Im Bereich der Methodenkompetenzen sollen innovativer Methoden im ausgewählten Bereich praktisch selbständig angewendet werden.</p> <p>Es sollen die Analyse-, Design- und Realisierungskompetenzen entwickelt werden, dabei die Fähigkeit, Probleme aus einem neuen und in der Entwicklung begriffenen Bereich des Softwareengineering zu formulieren, zu strukturieren und zu lösen.</p>				
3	<p>Inhalte</p> <p>Ausgewählte aktuelle Themen aus dem Bereich Softwareengineering und Behandlung forschungsnaher und praxisrelevanter Fragestellungen. Untersuchung von Methoden und Technologien des Software Engineerings in folgenden Gebieten:</p> <ul style="list-style-type: none"> Effizientes Requirements Engineering <ul style="list-style-type: none"> Tracing, Volere Modellgetriebene Softwareentwicklung <ul style="list-style-type: none"> MDSD, MOF, CIM, PIM, PSM, Modell, Meta-Modell, DSL Aspektororientierte Softwareentwicklung <ul style="list-style-type: none"> Crosscutting Concerns, Join Points, PointCuts, Advice, Aspect Event Driven Architecture <ul style="list-style-type: none"> JMS, ESB, CEP, DSMS Interaktive Web Applikationen <ul style="list-style-type: none"> MVC, MVP, MVVM Semantische Web Applikationen <ul style="list-style-type: none"> Open Data, Linked Data, RDF/S, OWL, SPARQL, RIF RuleML Agile Methoden und Prozesse <ul style="list-style-type: none"> TDD, ASD, Crystal, FDD, Kanban, BDD Software-Qualitätssicherung <ul style="list-style-type: none"> Metriken, Refactoring Projektmanagement in der Softwareentwicklung <ul style="list-style-type: none"> Teamorganisation, CMM, Aufwandsschätzung, Fortschrittskontrolle 				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Objektorientierte Programmierung, Datenbanken, Software Engineering				
6	Prüfungsvoraussetzungen keine				
7	Prüfungsformen Performanzprüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) keine				
10	Stellenwert der Note für die Endnote 10/90				

11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Jörg Brunsmann
12	Sonstige Informationen Literaturangaben: <ul style="list-style-type: none">• Volker Gruhn, Daniel Pieper, Carsten Röttgers. MDA. Effektives Softwareengineering mit UML2 und Eclipse. Springer Verlag, Berlin; 2006.• Opher Etzion, Peter Niblett: Event Processing in Action, 2010.• Russ Miles: AspectJ Cookbook, O'Reilly Media, 2005• Addy Osmani: "Developing Backbone.js Applications", O'Reilly Media, 2013• Toby Segaran, Colin Evans, Jamie Taylor: "Programming the Semantic Web", O'Reilly Media, 2009

Business Engineering und IT-Projektmanagement					
Kenn-Nr. 1.2	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Jedes SoSe	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 4 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Seminaristischer Unterricht 35	
2	Lernergebnisse (learning outcomes) / Kompetenzen Innovationen in der Informations- und Kommunikationstechnologie treiben die Um- und Restrukturierung komplexer Unternehmungen, indem sie neue Produktionsverfahren, Vertriebs- und Akquisewege, Verwaltungsstrukturen oder Businessprozesse ermöglichen. Diese betrieblichen Veränderungen gehen von reiner Automatisierung über neue Prozessmodelle bis hin zu echten Paradigmenwechseln. Die Implementierung solcher neuer Technologie und Unternehmensstrukturen muss unter Beteiligung des IT- und des Businessmanagements erfolgen. Hierzu dienen die in dieser Veranstaltung vermittelten Verfahren, Methoden und Instrumente, wie Enterprise Architecture, Servicemanagement, Projektmanagement- und (interkulturelle) Kommunikationskompetenzen. Die Studierenden sollen insbesondere eine ganzheitliche Betrachtung der Unternehmung im Rahmen des Business Engineerings anwenden.				
3	Inhalte Folgende Themen werden behandelt, eine inhaltliche Schwerpunktsetzung erfolgt jeweils anhand der aktuellen wissenschaftlichen und gesellschaftlichen Diskussion und aktuellen Forschungsprojekten: <ul style="list-style-type: none"> • Fortgeschrittene Aspekte des Projektmanagements • Methoden und Instrumente des IT-Projektmanagements • Projektmanagementframeworks und Vorgehensmodelle z.B. PMI, PRINCE2 • Enterprise Architecture Methodik • Enterprise Architecture Vorgehensmodelle z.B. TOGAF 9.1 • IT-Servicemanagement und Frameworks z.B. ITIL • Business Process Modeling und -Management • (Interkulturelle) Kommunikation im Change Management • Gesellschaftliche Implikationen von Unternehmensumgestaltungen 				
4	Lehrformen Seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Grundlagenkenntnisse im Projektmanagement				
6	Prüfungsvoraussetzungen keine				
7	Prüfungsformen Hausarbeit, Projektabgabe, Mündliche Prüfung, Klausur, Testat				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking				
12	Sonstige Informationen Literaturhinweise: o.A., ITIL Lifecycle Suite 2011, The Stationery Office Ltd, London 2011 o.A., Managing Successful Projects with PRINCE 2, The Stationery Office Ltd, London 2009 PMI (Hg.), A Guide to the Project Management Body of Knowledge, Newton Square, USA, 2008 Tiemeyer, E., Handbuch IT-Projektmanagement: Vorgehensmodelle, Managementinstrumente, Good Practices, München, 2010 The Open Group (Hg.), TOGAF Version 9.1, Reading, UK, 2011 Keller, W., IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur optimalen IT-Unterstützung, Heidelberg, 2012 Brocke, J.v., Rosemann, M. (Hgg.), Handbook on Business Process Management, Springer, 2011				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Architekturen, Modelle und Sprachen moderner Informationssysteme

Kenn-Nr. 1.3	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum	
2	Lernergebnisse (learning outcomes) / Kompetenzen Moderne Informationssysteme bedienen sich einer Vielzahl an Konzepten und Architekturen, um der immer größer werdenden Datenflut innerhalb des Unternehmens und an dessen Peripherie und der steigenden Komplexität der Software Herr zu werden. Ziel der Veranstaltung ist es, moderne Tendenzen und Konzepte von Informationssystemen aufzuzeigen. Die Studierenden lernen komplexe Architekturen und Modelle von Informationssystemen kennen. Die Studierenden kennen aktuelle technische Tendenzen moderner Informationssysteme und können sinnvolle Anwendungsszenarien aufzeigen. Sie wenden ihr Wissen in exemplarischen Implementierungen an und nutzen dabei aktuelle Technologien. Sie wenden sicher aktuelle Sprachen und Modellierungskonzepte an.				
3	Inhalte Folgende Themen werden behandelt, eine inhaltliche Schwerpunktsetzung erfolgt jeweils anhand der aktuellen wissenschaftlichen Diskussion und aktuellen Forschungsprojekten: <ul style="list-style-type: none"> • Verteilte Datenbanken und Cloud Data Management • Big Data – hochvolatile Datenmengen und Complex Event Processing • Business Intelligence • Objektrelationale Sprachkonzepte und ihre Implementierung • Semistrukturierte und unstrukturierte Datenbankinhalte • Datenbanken für spezielle Anwendungen • Architekturpattern wie beispielsweise <ul style="list-style-type: none"> ○ hierarchische Schichtenarchitekturen ○ (Client/Server, 3-tier, Peer-to-Peer), ○ Serviceorientierte Architekturen (SOA), ○ Webservices (UDDI, WSDL, SOAP, XML-RPC), ○ komponentenorientierte Architekturen (AUTOSAR) • Modellierung, Wiederverwendung und Weiterentwicklung von Architekturen • Domain-Driven Design • Architekturbeschreibungssprachen • Bewertung von Architekturen • Rolle des Softwarearchitekten 				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Grundkenntnisse in Datenbanksystemen (z.B. Besuch einer VL Datenbanken), Grundkenntnisse im Software Engineering (z.B. Besuch einer VL Software Engineering)				
6	Prüfungsvoraussetzungen keine				
7	Prüfungsformen Hausarbeit, Projektabgabe, Mündliche Prüfung, Klausur, Testat				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				

11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking
12	Sonstige Informationen Literaturhinweise: Friedland, A. et.al., N NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Hanser, München 2011 Fowler, M., Rice, D., Foemmel, M., Hieatt, E., Mee, R. und Stafford, R.: Patterns of Enterprise Application Architecture. Addison-Wesley, 2003. Thomas Erl: Service Oriented Architecture -- Concepts, Technology, and Design. Prentice Hall, 2006 Siedersleben, J: Moderne Softwarearchitektur; dpunkt.Verlag 2004 Starke, G.: Effektive Software Architekturen; 2. Aufl.; Hanser Verlag 2005

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Grafische Datenverarbeitung

Kenn-Nr. 1.4	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erlangen vertiefende Kenntnisse in den Bereichen geometrisches Modellieren mit Kurven und Flächen, Polygonale Netze, Bildsynthese und Visualisierung. Es wird davon jeweils ein Themenkomplex vertiefend behandelt. Die Studierenden sind in der Lage, grundlegende Forschungsarbeiten auf dem Gebiet der Computergrafik zu verstehen und einzuordnen.				
3	Inhalte In der Computergrafik werden aus Daten Bilder erzeugt. Dazu zählen die Teilbereiche geometrische Modellierung und Bildsynthese sowie Visualisierung. Im Seminaristischen Unterricht und im Praktikum Computergrafik wird eines der folgenden Themen vertiefend behandelt: Geometrisches Modellierung: Kurven und Flächen: Bezier-Kurven und Flächen, B-Spline-Kurven und Flächen, Unterteilungsflächen. Polygonale Netze: Datenstrukturen, Modellieren mit Netzen Bildsynthese und Visualisierung: Menschliche Wahrnehmung, Beleuchtung und Schattierung, Mapping Techniken. Visualisierungsmethoden, Virtual Reality Systeme. Weitere aktuelle Verfahren und Methoden der Computergrafik werden sofern möglich im Rahmen von begleitenden Industriekooperationen und Forschungsprojekten erlernt. Die behandelnden Verfahren und Algorithmen werden im Praktikum implementiert und intensiv untersucht.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Grundkenntnisse Computergrafik (z.B. Teilnahme an der Vorlesung Computergrafik Bachelor)				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller				
12	Sonstige Informationen Literaturangaben: <ul style="list-style-type: none"> • Bender M., Brill, M.: Computergrafik, 2. Auflage, Hanser Verlag, 2005. http://www.vis-lab.de • Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition. • Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics – Principles and Practice, Addison-Wesley 				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Methoden der praktischen Optimierung

Kenn-Nr. 1.5	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden erkennen, wenn ein Problem für ein Optimierungsverfahren zugänglich ist, und können es als ein entsprechendes Optimierungsproblem formulieren. Sie sind mit den wesentlichen Verfahren zur Lösung dieser Probleme vertraut und können diese Verfahren einschätzen, anwenden und umsetzen.				
3	Inhalte Grundlagen und Algorithmen für - Lineare Optimierung - Diskrete Optimierung - Nichtlineare Optimierung (mit/ohne Nebenbedingungen) - Multikriterielle Optimierung sowie Anwendungen der Optimierung. Aus Sicht der Informatik liegt der Schwerpunkt auf Algorithmen bzw. numerischen Lösungsverfahren.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: Gute Kenntnisse in Linearer Algebra, Analysis, Numerik				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: <ul style="list-style-type: none"> • U. Diwekar, Introduction to Applied Optimization, Springer 2003 • J. Nocedal and S. Wright, Numerical Optimization, Springer, 2006 				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Compilerbau

Kenn-Nr. 1.6	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Teilnehmer erlangen vertiefte Grundkenntnisse im Compilerbau. Sie sind mit dem Aufbau von Compilern und den Phasen der Übersetzung vertraut und können diese Kenntnisse zur Erstellung eigener Sprachen und Compiler anwenden. Ein besonderes Augenmerk liegt auf der Erstellung und Anwendung von Domainspezifischen Sprachen (DSL). Die behandelten Verfahren werden im Praktikum bei der Erstellung eines kleinen Compilers angewendet.				
3	Inhalte Auswahl an Themen zur Wissensvermittlung: - Überblick über Programmierparadigmen und -konzepte - Einstieg Compilerbau <ul style="list-style-type: none"> ■ Formale Sprachen, Syntax und Grammatik ■ Scanner und Parser (LR(k), LL(k)), Generatoren (z.B. Flex, Bison) ■ Codetransformation und Codegenerierung ■ Parser-/Compilergeneratoren (z.B. ANTLR) - Domainspezifische Sprachen (intern, extern), Implementierung mit XText - Modellierung mit Ecore und EMF				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Theoretische Informatik, Programmiermethoden, Software Engineering				
6	Prüfungsvoraussetzungen keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips				
12	Sonstige Informationen Literatur <ul style="list-style-type: none"> • Aho, Lam, Sethi, Ullman: "Compilers: Principles, Techniques, and Tools", Addison Wesley, 2006 • Torczon, Cooper: "Engineering a Compiler", Academic Press, 2011 • Parr, T.: "Language Implementation Pattern", Pragmatic Programmers, 2010 • Voelter, M.: "DSL Engineering: Designing, Implementing and Using Domain-Specific Languages", CreateSpace Independent Publishing Platform, 2013 • Ghosh, D.: "DSLs in Action", Manning, 2011 • Gronback, R.C.: "Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit", Addison-Wesley Longman, 2009 • Steinberg et.al.: "EMF: Eclipse Modeling Framework", Addison-Wesley Longman, 2008 				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Moderne Datenbanksysteme					
Kenn-Nr. 1.7	Work-load 150 h	Credits 5 cps	Studiensemester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Getrieben durch extreme Anforderungen im Bereich Big Data hat es eine stürmische Entwicklung im Bereich der Datenbanksysteme jenseits des klassischen RDBMS gegeben. Die Studierenden erwerben einen Überblick über Theorie, Architekturen, Implementierungstechniken, Sprachen und Anwendungen neuer Datenbanksysteme. Sie fällen begründete Entscheidungen für die Anwendung neuer DBMS. Sie können neue DBMS installieren und administrieren und dabei die Anforderungen der Anwendungssoftware analysieren und umsetzen. Sie können die Performanz von Datenbanken analysieren und durch geeignete technische Maßnahmen steigern. Sie befüllen DBMS mit großen Mengen an Beispieldaten und sprechen sie aus selbst implementierten Applikationen an.				
3	Inhalte Folgende Themen werden behandelt, eine inhaltliche Schwerpunktsetzung erfolgt jeweils anhand der aktuellen wissenschaftlichen Diskussion und aktuellen Forschungsprojekten: <ul style="list-style-type: none"> • Semistrukturierte und unstrukturierte Datenbankinhalte • Datenbanken für moderne Anwendungen • Datenbanken für extreme Anwendungsbeispiele • NoSQL DBMS diverser Flavours • DBMS mit heterogenen Architekturprinzipien • APIs moderner DBMS • Sprachen und Entwicklungsumgebungen für moderne DBMS • Parametrisierung und Optimierung moderner DBMS • Messung und Beurteilung der Performanz moderner und klassischer DBMS • DaaS 				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Vertiefte Kenntnisse in Datenbanksystemen (z.B. Besuch einer VL Datenbanken und einer Vertiefungsveranstaltung z.B. Implementierungstechniken von Datenbanken)				
6	Prüfungsvoraussetzungen keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking				
12	Sonstige Informationen Literaturhinweise: Zeitschriften und Proceedings zum Thema Friedland, A. et.al., N NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Hanser, München 2011 Redmond, E; Wilson, J.R.: Seven Databases in Seven Weeks – A Guide to Modern Databases and the NoSQL Movement. Pragmatic Programmers, o.O. 2012 Tiwari, S.: Professional NoSQL. Wiley, Indianapolis, 2011				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Effiziente Algorithmen

Kenn-Nr. 1.8	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen wichtige komplexe Algorithmen aus dem Gebiet der Informatik, die sich für eine Mehrzahl von Anwendungen eignen bzw. die Informatik stark geprägt haben. Sie sind in der Lage, die erlernten Algorithmen hinsichtlich ihres Einsatzes zu bewerten und selbstständig anzuwenden.				
3	Inhalte Beispielsweise: - Algorithmen zur Berechnung, Kodierung, Verschlüsselung - Grundlegende Algorithmen des Data Science, Machine Learning, Pattern Recognition - Simulationsalgorithmen - Auswahl von parallelen und verteilten Algorithmen - Fortgeschrittene Algorithmen für Graphen - gerade aktuelle Themen (z.B. Algorithmen im Hochfrequenz-Trading)				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: Lineare Algebra, Analysis, Algorithmen und Datenstrukturen				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König, Prof. Dr. Carsten Gips				
12	Sonstige Informationen Literatur: wird in der Veranstaltung bekannt gegeben.				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Innovationsmanagement und Technologietransfer

Kenn-Nr. 1.9	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst- studium 90 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die grundlegenden Mechanismen von Innovationen unterliegenden Prozessen. Sie sind in der Lage, proprietäres Wissen einzuordnen und kennen die Möglichkeit der Absicherung dieses Wissens. Sie können eigene Ideen in Geschäftspläne fassen.				
3	Inhalte <ul style="list-style-type: none"> - Wissen und Wissensmanagement - Kreativitätstechniken - Kontinuierliche und disruptive Innovation - Intellectual Property - Geschäftsplanung - Strategie und Markteintritt - Entrepreneurship 				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: -				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Hausarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Martin Hoffmann, Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: wird in der Veranstaltung bekannt gegeben.				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Mobile Anwendungen

Kenn-Nr. 1.10	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst- studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Teilnehmer erlangen fortgeschrittene Kompetenzen auf dem Gebiet der Softwareentwicklung für mobile Geräte. Sie sind mit verbreiteten Hardwareplattformen und mobilen Betriebssystemen vertraut und können problemorientiert passende Frameworks zur Softwareentwicklung auswählen und einsetzen. In praktischen Beispielen werden die behandelten Verfahren eingeübt.				
3	Inhalte <ul style="list-style-type: none"> • Grundlagen der Softwareentwicklung für mobile Systeme (Status Quo, Forschungsgegenstände, Trends) • Überblick über aktuelle Hardware und Betriebssysteme • Aktuelle Software-Frameworks für mobile Anwendungen • Entwurf von mobilen Systemen (u.a. Beschreibung mobiler Architekturen) • Implementierung von mobilen Systemen • Cross Plattform Development • Test von mobilen Softwaresystemen • Ressourcenmanagement in mobilen Systemen • Sicherheit und Privatsphäre • Energieaspekte • Datenspeicherung • Sensorik und Kontext (u.a. Geo-Lokalisierung) 				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich:				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Martin Hoffmann, Prof. Dr. Matthias König				
12	Sonstige Informationen				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Operations Research

Kenn-Nr. 1.11	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
					60 15
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können Probleme des Operations Research formulieren und mittels der erlernten Verfahren lösen. Außerdem sind sie mit der Umsetzung der Verfahren dahingehend vertraut, dass sie diese in einer Programmiersprache umsetzen können.				
3	Inhalte Beispielsweise: - Grundlagen und Anwendungen des Operations Research - Netzplantechnik - Lineare Programmierung - Warteschlangentheorie - Graphentheorie - Kombinatorische und dynamische Optimierung - Simulation				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: Gute Kenntnisse in Algebra, Analysis, Numerik, Stochastik, Optimierung				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: • Domschke/Drexel, Einführung in Operations Research, Springer, 2005				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Theoretische Kryptographie

Kenn-Nr. 1.12	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst- studium 90 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Diese Veranstaltung macht die Studierenden mit den grundlegenden Konzepten und Methoden der theoretischen Kryptographie vertraut. Die Studierenden verfügen über ein fundiertes Verständnis der komplexitätstheoretischen Grundlagen der Kryptographie, können entsprechende Sicherheitsannahmen und –garantien interpretieren und darauf basierende Sicherheitsbeweise überprüfen. Weiterhin kennen die Studierende die wichtigsten kryptographischen Grundprimitive und Protokolle und können diese zu komplexeren Protokollen kombinieren.				
3	Inhalte Im Seminaristischen Unterricht werden theoretische Grundlagen und Konzepte der modernen Kryptographie eingeführt. Dazu gehören unter anderem Einwegfunktionen; Hardcore-Bits; Pseudozufall; Commitment Schemes; Zero-Knowledge Beweise; Simulationsbasierte Sicherheit; Mehrparteienberechnungen;				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: Vorkenntnisse im Bereich Kryptographie sind hilfreich, aber nicht zwingend notwendig. Theoretische Informatik				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christoph Thiel				
12	Sonstige Informationen Literatur: <ul style="list-style-type: none"> • O. Goldreich, Foundations of cryptography 1 + 2, Cambridge University Press, 2009. • J. Katz, Introduction to Modern Cryptography: Principles and Protocols (Chapman & Hall/CRC Cryptography and Network Security Series), 2007. 				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Fortgeschrittene Methoden der Künstlichen Intelligenz

Kenn-Nr. 1.13	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum	
	2 SWS 2 SWS			60 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden lernen fortgeschrittene Methoden und Verfahren der modernen Künstlichen Intelligenz kennen. Mit der erfolgreichen Teilnahme an der Vorlesung und den Praktika werden Kompetenzen zur Auswahl und Anwendung von aktuellen Algorithmen der KI für eine konkrete Problemstellung erworben. Die Teilnehmer sind in der Lage, die erlernten Methoden auf andere Bereiche und Problemstellungen zu übertragen und anzuwenden.				
3	Inhalte Eine Auswahl aus folgenden Themen wird behandelt, eine inhaltliche Schwerpunktsetzung erfolgt jeweils anhand der aktuellen wissenschaftlichen Diskussion und aktuellen Forschungsprojekten. Auswahl an Themen zur Wissensvermittlung: <ul style="list-style-type: none"> - KI und Spiele: Taktik, Strategie, Lernen - Robotik, mobile Roboter, Navigation im Raum, autonome mobile Systeme - Unschärfe Textsuche, Ähnlichkeiten zw. Texten, Empfehlungssysteme - Maschinelles Lernen, Neuronale Informationsverarbeitung, Probabilistisches Schliessen - Logik und Kalküle: Prädikatenlogik und Resolution, Fuzzylogik, nichtmonotone Logik, temporale Logik - Wissensrepräsentation und Inferenz, Semantische Netze - Planen - Numerische/Symbolische Beschränkung und Propagierung (Constraint Satisfaction Problems) - Genetische und Evolutionäre Algorithmen - Natürliche Sprache, Spracherkennung, Syntaktische Analyse, Semantik - Bildverarbeitung und Mustererkennung 				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Theoretische Informatik, Algorithmen und Datenstrukturen				
6	Prüfungsvoraussetzungen keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips				
12	Sonstige Informationen: Literatur <ul style="list-style-type: none"> • Russell, Norvig: "Artificial Intelligence" • Bishop: "Neural Networks for Pattern Recognition" • Bishop: "Pattern Recognition and Machine Learning" • Thrun: "Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)" • Millington, Funge: "Artificial Intelligence for Games" 				

- | | |
|--|---|
| | <ul style="list-style-type: none">• Mitchell: "Machine Learning"• Haykin, S.: "Neural Networks and Learning Machines: A Comprehensive Foundation", Prentice Hall, 2008 |
|--|---|

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Web-Application-Security

Kenn-Nr. 1.14	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesungen Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Vorlesung Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden besitzen Kenntnisse zu prinzipiellen Angriffsszenarien und typischen Schwachstellen von Web-Applikationen und Web-Servern. Sie haben ein solides Verständnis der grundlegenden Vorgehensweisen bei Planung und Durchführung von Penetrationstests von Web-Anwendungen und erste praktische Erfahrungen bei der Durchführung von Penetrationstests mit manuellen und halbautomatischen Werkzeugen. Die Studierenden können das Sicherheitsniveau von Web-Servern und Web-Anwendungen analysieren und bewerten und sind in der Lage sichere Web-Server und Web-Anwendungen zu entwerfen.				
3	Inhalte <ul style="list-style-type: none"> • Funktionsweise und Architektur von Webanwendungen • Analyse von Angriffsszenarien und typischen Schwachstellen von Webanwendungen (z.B. OWASP Top 10) • Methoden und Werkzeuge zum Penetration Testing • Vorgehensmodelle für Sicherheitstests (z.B. OSSTMM 3, usw.) • Sicherung und Härtung von Webanwendungen • (automatische) Erkennung von Sicherheitslücken von Webanwendungen • Web Application Firewalls (WAF) • Sicherheitsbewertung • Entwurf sicherer Webanwendungen 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich:				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christoph Thiel				
12	Sonstige Informationen Literatur: <ul style="list-style-type: none"> • Frank Neugebauer: Penetration Testing mit Metasploit: Eine praktische Einführung, dpunkt.verlag GmbH; Auflage: 2., akt. u. erw. Aufl., ISBN-10: 3898648206, 2012 • Mike Shema: Hacking Web Apps: Detecting and Preventing Web Application Security Problems, Syngress Media, ISBN-10: 159749951X, 2012 				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Funktionale Programmierung

Kenn-Nr. 1.15	Work-load 150 h	Credits 5 cps	Studiense- mester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesungen Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die funktionale Programmierung ist neben der objektorientierten Programmierung und der verteilten/nebenläufigen Programmierung ein wichtiges Programmierparadigma. Konzepte aus der funktionalen Programmierung wie beispielsweise Lambda-Ausdrücke halten schrittweise Einzug in moderne Sprachen wie Java 8 und C#. Die Veranstaltung führt die Konzepte der funktionalen Programmierung ein und zeigt die Umsetzung in den Beispielsprachen Haskell und Scala. Die Studierenden beherrschen wichtige Konzepte der funktionalen Programmierung und können diese anhand der Sprachen Haskell und Scala anwenden. Sie erkennen funktionale Konzepte in anderen modernen Programmiersprachen und können diese ergebnisorientiert anwenden.				
3	Inhalte Auswahl der Themen: <ul style="list-style-type: none"> • Funktionen und Operatoren • Lambda-Notation • Funktionen höherer Ordnung: map, filter, reduce, zip • Funktionskomposition und Currying • Datenstrukturen • Typen und Typklassen, Polymorphie • Funktoren und Monaden • Auswertungsstrategien • Modularisierung und Schnittstellen • Berechenbarkeit und Lambda-Kalkül • Einführung in die Programmiersprachen Haskell und Scala 				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Spezielle Methoden der Programmierung				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips				
12	Sonstige Informationen Literatur: <ul style="list-style-type: none"> • Pepper, Hofstedt: "Funktionale Programmierung", Springer, 2006 • Jeuring, Meijer: "Advanced Functional Programming", Springer, 1995 • Block, Neumann: "Haskell Intensivkurs", Springer, 2011 • Horstmann, C.: "Scala for the Impatient", Addison Wesley, 2012 • Odersky, M.: "Programming in Scala", Artima, 2011 				

Wahlfach aus der Liste „Fortgeschrittene Methoden der Informatik“ Public Key Infrastructure						
Kenn-Nr. 1.16	Work-load 150 h	Credits 5 cps	Studiensemester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester	
1	Lehrveranstaltungen Vorlesungen Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Vorlesung Praktikum	60 15
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Unter dem Begriff „Public Key Infrastruktur“ (PKI) versteht man die gesamte Infrastruktur, die notwendig ist, um Schlüssel für ein asymmetrisches Verschlüsselungsverfahren zu erzeugen, an Nutzer auf sicherem Weg weiterzugeben, zu verwalten und ggf. zu widerrufen. Somit basieren die meisten modernen Sicherheitslösungen auf dem Einsatz einer oder mehrerer verschiedener PKIs.</p> <p>Die Studierenden sind in der Lage, die Funktionsweise der grundlegenden Verfahren einer PKI zu verstehen und diese in einem praktischen Feld einzusetzen. Sie haben ein solides Verständnis der notwendigen Vorgehensweisen bei Planung und Konzeption von PKIs und erste praktische Erfahrungen bei deren Implementierung. Sie haben ein gutes Verständnis von möglichen Angriffen und geeigneten Gegenmaßnahmen im Bereich von PKI-orientierten Sicherheitslösungen.</p>					
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Sicherheitsziele • Public Key Kryptographie (Verschlüsselungs-, Signatur-, Hashverfahren) • Zertifikate (X.509, PGP, Attributzertifikate) • Vertrauensmodelle (Web of Trust, Hierarchische Vertrauensmodelle) • Private Schlüssel (Software-, Hardware-Token) • Revozierung (Certificate Revocation Lists, OCSP) • Policies (Certificate Policy, Certificate Practice Statement) • Gültigkeitsmodelle (Schalen-, Kettenmodell) • Zertifizierungspfade und -Protokolle (LDAP, SCVP, Long Term Archiving) • Trust Center (Registration Authority, Certification Authority, Certificate Management Authority) • Standards und rechtliche Rahmenbedingungen 					
4	Lehrformen Vorlesung, Praktikum					
5	Teilnahmevoraussetzungen Formal: - Inhaltlich:					
6	Prüfungsvoraussetzungen Keine					
7	Prüfungsformen Klausur, mündliche Prüfung					
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung					
9	Verwendung des Moduls (in anderen Studiengängen) Keine					
10	Stellenwert der Note für die Endnote 5/90					
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christoph Thiel					
12	<p>Sonstige Informationen</p> <p>Literatur:</p> <ul style="list-style-type: none"> • J. Buchmann, E. Karatsiolis, and A. Wiesmaier. "Introduction to Public Key Infrastructures", Springer-Verlag Berlin Heidelberg, 2013. ISBN: 978-3-642-40656-0 (Print) 978-3-642-40657-7 (Online) • J. Buchmann, "Einführung in die Kryptographie", ISBN 3-540-41283-2 					

- | |
|--|
| <ul style="list-style-type: none">• C. Adams / S. Lloyd, "Understanding Public-Key Infrastructure", ISBN 1-57870-166-X• Tom Austin, "PKI / A Wiley Tech Brief", ISBN 0-471-35380-9• R. Housley / T. Polk, "Planning for PKI", ISBN 0-471-39702-4• A. Nash / W. Duane / C. Joseph/ D. Brink, "PKI Implementing and Managing E-Security", ISBN 0-007-213123-3• Henk C.A. van Tilborg, "Encyclopedia of Cryptography and Security", ISBN-13: 978-0387234731 |
|--|

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Computer Vision					
Kenn-Nr. 1.17	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS /90 h	Selbst-studium 210 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Das Ziel ist die Förderung der Selbstständigkeit und praktischen Problemlösungskompetenz sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten. Das vom Dozenten gestellte Projekt dient der Wissensvertiefung im Bereich des maschinellen Sehens / Computer Vision und als praktische Erfahrung in der Projektplanung und -realisierung eines Forschungsprototyps.				
3	Inhalte Der Seminaristische Unterricht behandelt Verfahren Computer Vision, welche auf den Projektinhalt abgestimmt sind. Eine Analyse der Verfahren und deren Umsetzungen im Projektkontext führen die Studierenden durch. Beispielhafte Auswahl der Themen: <ul style="list-style-type: none"> • Videoinhaltsanalyse • Bewegungsanalyse • Stereobildauswertung • Structure from Motion • 3D Rekonstruktion • Image Retrieval • Wissensbasierte Bildanalyse Ablauf der Projekte Weitgehend selbstständige Bearbeitung einer komplexeren Aufgabenstellung im Rahmen eines Forschungs- und Entwicklungsprojekts im Team, welches auch in Kooperation mit F&E-Abteilungen von Firmen bearbeitet werden kann. In der Regel bestehen die Projektgruppen aus 3-5 Studierenden, die sich frei zusammen finden und einen Projektleiter aus ihren Reihen wählen und nach mit dem Dozenten vereinbarten Vorgehensmodellen entwickeln. Der Dozent definiert die Zielsetzung und führt einen regelmäßigen Diskurs über den Fortgang des Projekts. Er vereinbart außerdem mit dem Team Meilensteine sowie Kommunikations- und Kooperationsformen.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: Lineare Algebra, Analysis, Mustererkennung und Bildverarbeitung, Programmierung in C++				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandenes Praktikum mit Testat, Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: wird in der Veranstaltung bekannt gegeben.				

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Visualisierung					
Kenn-Nr. 1.18	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 4 SWS	Kontaktzeit 6 SWS /90 h	Selbst-studium 210 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden haben nach dem Kurs einen Überblick über geeignete Verfahren und Methoden zur visuellen Problemlösung aktueller Fragestellungen in der Computergrafik und können die ausgewählten Algorithmen umsetzen. Die Studierenden erhalten Kenntnisse zur Implementierung von Softwareapplikationen zu Problemstellungen aktueller Forschungsprojekte der Computergrafik wie z.B. geometrische Modellierung, Visualisierung, Virtual Reality, Animation und (physikalische) Simulation, Mustererkennung und Bildverarbeitung, Human Computer Interaction sowie Fragestellungen aus interdisziplinären Forschungsprojekten mit Computergrafik als wissenschaftliche Anwendung. Die Studierenden bekommen Erfahrung in der Planung und Realisierung eines Forschungsprojektes in der Computergrafik und sind in der Lage, Forschungsarbeiten in dem behandelten Gebiet zu verstehen und einzuordnen.				
3	Inhalte Computergrafik ist der Bereich der Datenverarbeitung, der sich mit der Erfassung, Speicherung, Verarbeitung und Ausgabe graphischer Darstellungen befasst. Im Rahmen dieses Kurses werden aktuelle Visualisierungstechniken zur Darstellung und Lösung interdisziplinärer Probleme erlernt. Die theoretischen Grundlagen werden im Seminaristischen Unterricht erarbeitet. Im Praktikum werden Daten aus aktuellen Industrie- und Forschungsprojekten geeignet visualisiert. Dazu sind begleitende Industriekooperationen und Forschungsprojekte angestrebt. Die Fragestellungen sind schwerpunktmäßig in den folgenden Gebieten der Computergrafik angesiedelt: <ul style="list-style-type: none"> • Visualisierung von Bezier und B-Spline Flächen sowie Unterteilungsflächen. • Virtual Reality Anwendungen im low-cost Bereich. • Animation und graphische Simulation • Die computergrafisch spezifischen Anwendungen sind im Team zu bearbeiten. Der Praktikumssteil kann als Vorbereitung für die Masterarbeit und vertiefende Industriepraktika im Bereich Computergrafik angesehen werden. 				
4	Lehrformen Seminaristischer Unterricht / Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Grundkenntnisse Computergrafik (z.B. Teilnahme an der Vorlesung Computergrafik Bachelor)				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektanfertigung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Kerstin Müller				
12	Sonstige Informationen Literaturangaben: <ul style="list-style-type: none"> • Bender M., Brill, M.: Computergrafik, 2. Auflage, Hanser Verlag, 2005. http://www.vislab.de • Hearn D., Baker M.P.: Computer Graphics with OpenGL, Pearson International Edition. • Foley J., van Dam A., Feiner S., Hughes J.: Computer Graphics – Principles and Practice, Addison-Wesley 				

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Hardwarenahe Programmierung					
Kenn-Nr. 1.19	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS /90 h	Selbst- studium 210 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Das Ziel ist die Förderung der Selbstständigkeit und praktischen Problemlösungs-kompe- tenz sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten. Das vom Dozenten gestellte Projekt dient der Wissensvertiefung im Bereich der hardwarenahen Programmierung und als praktische Erfahrung in der Projektplanung und -realisierung ei- nes Forschungsprototyps.				
3	Inhalte Hardwarenahe Programmierung ist sowohl für den direkten Zugriff auf Hardware als auch die effiziente Nutzung dieser von großer Bedeutung. Dies gilt insbesondere für die Ent- wicklung von performanter Software für Embedded Systems und somit für eine Vielzahl von Anwendungen. Der Seminaristische Unterricht behandelt Verfahren der hardwarenahen Programmierung, welche auf den Projektinhalt abgestimmt sind. Eine Analyse der Verfahren und deren Um- setzungen im Projektkontext führen die Studierenden durch. Auswahl der Themen: <ul style="list-style-type: none"> • Embedded Linux • Videocompression • Echtzeitbildverarbeitung • Sensor-/Aktornetzwerke • Autonome Roboter • Mess- und Regelsysteme • Telematik Ablauf der Projekte Weitgehend selbstständige Bearbeitung einer komplexeren Aufgabenstellung im Rahmen eines Forschungs- und Entwicklungsprojekts im Team, welches auch in Kooperation mit F&E-Abteilungen von Firmen bearbeitet werden kann. In der Regel bestehen die Projekt- gruppen aus 3-5 Studierenden, die sich frei zusammen finden und einen Projektleiter aus ihren Reihen wählen und nach mit dem Dozenten vereinbarten Vorgehensmodellen entwi- ckeln. Der Dozent definiert die Zielsetzung und führt einen regelmäßigen Diskurs über den Fortgang des Projekts. Er vereinbart außerdem mit dem Team Meilensteine sowie Kommu- nikations- und Kooperationsformen.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: BSc Informatik Inhaltlich: Kenntnisse eingebetteter Systeme, Programmierung in C++				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektabgabe				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König				
12	Sonstige Informationen Literatur: wird in der Veranstaltung bekannt gegeben.				

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Design und Modellierung von Softwaresystemen					
Kenn-Nr. 1.20	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 4 SWS	Kontaktzeit 6 SWS /90 h	Selbst- studium 210 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Ziel der Veranstaltung ist die Problemlösungskompetenz der Studierenden so zu entwickeln, dass Sie technische Führungsaufgaben in großen Softwareentwicklungsprojekten übernehmen können. Besonderes Augenmerk liegt auf der Methodenkompetenz und hier besonders auf Methoden des selbständigen wissenschaftlichen Arbeitens. Das vom Dozenten gestellte Projekt dient zur Erarbeitung und der Anwendung wissenschaftlicher Methoden der Informatik insbesondere die erfindende Methode wissenschaftlicher Erkenntnisgewinnung.</p> <p>Die Studierenden sammeln praktische Erfahrungen in der Projektarbeit bei der wissenschaftlichen Recherche, beim Design und der Implementierung von Forschungsprototypen, der wissenschaftlichen Einordnung der Ergebnisse und der Präsentation und Darstellung der Ergebnisse in typischen Formaten der wissenschaftlichen Veröffentlichung.</p>				
3	<p>Inhalte</p> <p>Große Softwaresysteme sind Teil der Enterprise Architecture jedes mittleren und großen Unternehmens. Wissenschaftliche Ansätze zum Design und Modellierung solcher Systeme werden breit diskutiert und verändern sich kontinuierlich. Das Anwenden aktueller Methoden unter kritisch-wissenschaftlicher Betrachtungsweise ist wesentlicher Inhalt der Veranstaltung.</p> <p>Auswahl an Themen:</p> <ul style="list-style-type: none"> • Aktuelle Sprachen und Modelle des Softwaredesigns • Requirementsengineering und -modellierung • Architekturmodelle und Architekturmuster • Methoden und Vorgehensweisen der IT-Unternehmensarchitektur • Entwurf komplexer Informationssysteme – Strukturen für Daten und Information • Mobile Daten und Daten mobilisieren – Replikation, Verteilung • SaaS und DaaS <p>In Absprache mit dem Dozenten wählen die Studierenden Projektthemen in Gruppen aus und bearbeiten diese über ein Semester als Projektgruppe. Sie recherchieren den Stand der Technik und Wissenschaft, formulieren eine Forschungsfrage und Requirements und erstellen eine komplette fachliche und technische Modellierung und Dokumentation. Sie bedienen sich aktueller Projektmanagementmethoden und -tools. Sie implementieren ausgewählte Teile der Modellierung in funktionierende Software. Sie ordnen die Ergebnisse in die aktuelle wissenschaftliche Diskussion ein und erstellen eine Darstellung der Ergebnisse in typischen Formaten der wissenschaftlichen Veröffentlichung.</p>				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: - Inhaltlich: Grundkenntnisse Datenbanken (z.B. Besuch einer VL Datenbanken), Grundkenntnisse im Software Engineering (z.B. Besuch einer VL Software Engineering), Grundkenntnisse in wissenschaftlichem Arbeiten (z.B. Besuch eines Fachseminars)</p>				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektabgabe, Hausarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				

10	Stellenwert der Note für die Endnote 10/90
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking
12	Sonstige Informationen Literaturhinweise: Zeitschrift: Software and Systems Modeling, Springer, Heidelberg Starke, G; Effektive Softwarearchitekturen, Hanser, München, 2011 Autorenkollektiv, TOGAF Version 9.1, The Open Group, London, 2011

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Intelligente Systeme

Kenn-Nr. 1.21	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 4 SWS		Kontaktzeit 6 SWS /90 h	Selbst-studium 210 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15
2	Lernergebnisse (learning outcomes) / Kompetenzen Ziel ist die Förderung der Selbstständigkeit und praktischen Problemlösungskompetenz sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten. Das vom Dozenten gestellte Projekt dient der Wissensvertiefung im Bereich der Künstlichen Intelligenz und als praktische Erfahrung in der Projektplanung und -realisierung eines Forschungsprototyps.				
3	Inhalte Methoden der künstlichen Intelligenz finden zunehmend breiteren Einsatz in der Anwendungsentwicklung, bei Optimierungsfragestellungen, automatisierten Problemlösungen und in der Automatisierung. Spezielle Methoden der künstlichen Intelligenz sollen je nach Projektinhalten und Anwendungsfeld in der Veranstaltung vom Dozenten vermittelt werden, durch die Studierenden analysiert und in den projektspezifischen Implementierungen zum Einsatz kommen. Auswahl an Themen zur Wissensvermittlung: <ul style="list-style-type: none"> • Überblick über die Methoden der Künstlichen Intelligenz • Wissenrepräsentation und Inferenz, Semantische Netze • Problemlösen: Generiere-und-Teste-Verfahren, Suchverfahren • Prädikatenlogik, Regeln und Regelverarbeitung, Wissensmodellierung (z.B. mit Prolog) • Numerische/Symbolische Beschränkung und Propagierung • Genetische und Evolutionäre Algorithmen • Maschinelles Lernen • Neuronale Informationsverarbeitung, probabilistisches Schliessen • Planen • Natürliche Sprache, Spracherkennung, Syntaktische Analyse, Semantik • Bildverarbeitung und Mustererkennung • Robotik, mobile Roboter, Navigation im Raum • KI und Spiele: Taktik, Strategie, Lernen Ablauf der Projekte: Weitgehend selbstständige Bearbeitung einer komplexeren Aufgabenstellung im Rahmen eines Forschungs- und Entwicklungsprojekts im Team, welches auch in Kooperation mit Forschungs- und Entwicklungsabteilungen von Firmen bearbeitet werden kann. In der Regel bestehen die Projektgruppen aus 3-5 Studierenden, die sich frei zusammen finden und einen Projektleiter aus ihren Reihen wählen und nach mit dem Dozenten vereinbarten Vorgehensmodellen entwickeln. Der Dozent definiert die Zielsetzung und führt einen regelmäßigen Diskurs über den Fortgang des Projekts. Er vereinbart außerdem mit dem Team Meilensteine sowie Kommunikations- und Kooperationsformen.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Programmieren in Java, C++ und Python, Theoretische Informatik, Algorithmen und Datenstrukturen, Grundlagen der Webentwicklung, Softwaretechnik, Datenbanken bzw. der Mobilien Applikationsentwicklung				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektabgabe				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) Keine
10	Stellenwert der Note für die Endnote 10/90
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Grit Behrens, Prof. Dr. Carsten Gips
12	Sonstige Informationen Russell, Norvig: "Artificial Intelligence" Bishop: "Neural Networks for Pattern Recognition" Bishop: "Pattern Recognition and Machine Learning" Thrun: "Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)" Millington, Funge: "Artificial Intelligence for Games" Mitchell: "Machine Learning" Luger: "Artificial Intelligence"

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Systemsicherheit

Kenn-Nr. 1.22	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	Kontaktzeit 6 SWS /90 h	Selbst-studium 210 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum	
	2 SWS 4 SWS			60 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden besitzen einen Überblick über Strategien und Konzepte bzgl Sicherheit und den Schutzbedarf von Softwaresystemen, können Methoden bewerten und sind in der Lage technische Schutzmaßnahmen in der Praxis umzusetzen. Die Studierende können sicherheitskritische Anwendungen entwickeln und kennen dazu die Prinzipien der sicheren Programmierung.				
3	Inhalte Die Studierenden lernen im Seminaristischen Unterricht und im Praktikum den Entstehungsprozess sicherheitskritischer Anwendungen kennen. Dazu werden Sicherheitsanforderungen analysiert, ebenso wird eine Risiko- und Bedrohungsanalyse durchgeführt. Sicherheitsrichtlinien werden erlernt und eine Sicherheitsarchitektur erarbeitet. Grundlegende Sicherheitsmechanismen sowie Zugriffs- und Informationsfluß-Kontrollmodelle werden eingeführt, weiterhin Methoden zur Validierung und Bewertung sicherer Anwendungen. Die Studierenden erhalten einen Überblick über IT-Sicherheitsmanagement-Konzepte und das IT-Grundschutzkonzept des BSI. - IT-Sicherheitsstandards für Softwareprodukte und IT Infrastrukturen - Schwachstellenanalyse - Sicherheitskonzepte und IT-Sicherheitsmanagement - IT-Sicherheitsarchitekturen für Softwaresysteme - Aufbau und Komponenten einer IT-Sicherheitsarchitektur - Untersuchung aktueller Softwaresysteme Im Praktikum wird insbesondere auf die sichere Software-Entwicklung eingegangen und an einem praktischen Beispiel geübt. Die grundlegenden Angriffs-Vektoren von Hackern, Crackern und Schadsoftware-Autoren werden erläutert und am Praktikumsbeispiel getestet. Dabei soll sichtbar werden, wie unsichere Programmierung Verwundbarkeiten eröffnet und wie dies durch einen sicheren Software-Entwurfs und sicherere Programmierung zu verhindern ist. Als praxisrelevantes Beispiel wird der "Security Development Cycle" von Microsoft betrachtet sowie Methoden, welche die Güte solcher Prozesse messen und verbessern.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich:				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christoph Thiel				
12	Sonstige Informationen Literatur:				

- | | |
|--|--|
| | <ul style="list-style-type: none">• Claudia Eckert: IT-Sicherheit, 6. Auflage, Oldenbourg, 2009• Ross Anderson: Security Engineering, Wiley & Sons, 2008• Jack Koziol et.al.: The Shellcoder's Handbook, Wiley & Sons, 2007• Michael Howard et.al.: The Security Development Lifecycle, Microsoft Press, 2009 |
|--|--|

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Data Science

Kenn-Nr. 1.23	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	Kontaktzeit 6 SWS /90 h	Selbst-studium 210 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Getrieben durch extreme Anforderungen im Bereich Big Data, gibt es einen Trend zur Konsolidierung von Methoden und Erkenntnissen verschiedener Disziplinen der Informatik zu einer vereinheitlichten Data Science. Die Studierenden erarbeiten sich anhand eines komplexen Projektes aus dem Bereich Analyse und Verarbeitung großer Datenmengen einen wissenschaftlichen Zugang zu den Bereichen der Data Science.</p> <p>Das vom Dozenten gestellte Projekt dient zur Erarbeitung und der Anwendung wissenschaftlicher Methoden der Informatik insbesondere die erfindende Methode wissenschaftlicher Erkenntnisgewinnung, mathematischer und statistischer Methoden zur Datenanalyse und empirischer Forschung, insb. experimentelle Methoden zur Performanzmessung. Die Studierenden sammeln praktische Erfahrungen in der Projektarbeit bei der wissenschaftlichen Recherche, beim Design und der Implementierung von Forschungsprototypen, der wissenschaftlichen Einordnung der Ergebnisse und der Präsentation und Darstellung der Ergebnisse in typischen Formaten der wissenschaftlichen Veröffentlichung.</p>				
3	<p>Inhalte</p> <p>Data Science verwendet Methoden und Erkenntnisse aus verschiedenen Bereichen der Informatik. Das Anwenden solcher Methoden unter kritisch-wissenschaftlicher Betrachtungsweise ist wesentlicher Inhalt der Veranstaltung.</p> <p>Auswahl an Themen:</p> <ul style="list-style-type: none"> • Cloud Computing • Moderne Datenbankarchitekturen • Datenanalyse • Maschinelles Lernen • Natural Language Processing • Datenvisualisierung • Information Retrieval • Knowledge Discovery • Data Mining • Bad Data und Data Cleansing <p>In Absprache mit dem Dozenten wählen die Studierenden Projektthemen aus dem Bereich Big Data/Data Science in Gruppen aus und bearbeiten diese über ein Semester als Projektgruppe. Sie recherchieren den Stand der Technik und Wissenschaft, formulieren eine Forschungsfrage und erarbeiten das erforderliche Skillset. Sie bedienen sich aktueller Projektmanagementmethoden und -tools. Sie implementieren ausgewählte Teile der Modellierung in funktionierende Software. Sie ordnen die Ergebnisse in die aktuelle wissenschaftliche Diskussion ein und erstellen eine Darstellung der Ergebnisse in typischen Formaten der wissenschaftlichen Veröffentlichung.</p>				
4	<p>Lehrformen</p> <p>Seminaristischer Unterricht, Praktikum</p>				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: - Inhaltlich: Vertiefte Kenntnisse in Datenbanksystemen (z.B. Besuch einer VL Datenbanken und einer Vertiefungsveranstaltung z.B. Implementierungstechniken von Datenbanken), Grundkenntnisse in wissenschaftlichem Arbeiten (z.B. Besuch eines Fachseminars)</p>				
6	<p>Prüfungsvoraussetzungen</p> <p>Keine</p>				
7	<p>Prüfungsformen</p> <p>Projektabgabe, Hausarbeit</p>				

8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) Keine
10	Stellenwert der Note für die Endnote 10/90
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking
12	Sonstige Informationen Literaturhinweise: Zeitschriften und Proceedings zum Thema Friedland, A. et.al., N NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, Hanser, München 2011 O'Neill, C.; Schutt, R.: Doing Data Science. O'Reilly, Cambridge USA, 2013 McCallum, Q.E.: Bad Data Handbook. O'Reilly, Cambridge USA, 2012 McKinney, W.: Python for Data Analysis. O'Reilly, Cambridge USA, 2013 Witten, I.H. et.al.: Data Mining. Elsevier, Burlington USA, 2011

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Distributed Organic Computing Systems					
Kenn-Nr. 1.24	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS /90 h	Selbst- studium 210 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden vertiefen ihr Wissen im Bereich der Verteilten Systeme und erwerben fundierte Kenntnisse in einem oder mehreren der folgenden Teilbereiche: <ul style="list-style-type: none"> • Konzepte und Anwendungen von Cloud-Computing bzw. Peer-to-Peer-Systemen als Grundlage komplexer Internet basierter Infrastrukturen • Einsatzgebiete und Funktionsweise von ad-hoc und Sensornetzen und Unterschiede zwischen traditionellen Rechnernetzen und infrastrukturlosen Kommunikationsnetzen. Insbesondere werden aktuelle Erkenntnisse aus dem Forschungsgebiet Organic Computing, basierend auf grundlegenden Konzepten naturanaloger Algorithmen und der Funktionsweise selbstorganisierender verteilter Systeme, betrachtet. Die Studierenden erarbeiten eigene Ansätze für zukünftige Systemarchitekturen. Sie sammeln praktische Erfahrungen in der Projektarbeit, bei der wissenschaftlichen Recherche, beim Design und der Implementierung von Systemarchitekturen, der anschließenden Bewertung der Ergebnisse (unter technischen und wissenschaftlichen Aspekten) und der Präsentation und Darstellung der Ergebnisse.				
3	Inhalte Wissenschaftliche Vorarbeiten im Bereich Cloud Computing und Peer-to-Peer-Systeme liefern Ansätze zur Organisation von komplexen abstrahierten IT-Infrastrukturen, die dynamisch an wechselnde Nutzungsbedingungen angepasst werden können und Dienste auf verschiedenen Ebenen zur Verfügung stellen, z.B. Rechenkapazität, Datenspeicher, Netzkapazitäten und Softwaredienste. Als Sonderfälle werden ad-hoc und Sensornetze untersucht. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. Darüber hinaus werden Konzepte aus dem Bereich der Systemarchitekturen und Ansätze aus dem Bereich selbstorganisierender Algorithmen dargestellt und bewertet. Das Forschungsgebiet Organic Computing bietet aktuelle Ansätze zur Beherrschung von hoher Komplexität in technischen Systemen. Ausgehend von der Definition des Forschungsgebietes Organic Computing und seiner allgemeinen Zielsetzung werden insbesondere Konzepte und Mechanismen aus der Natur in technische Anwendungen und Algorithmen überführt. Dazu wird ein Verständnis für Probleme bei der Entwicklung komplexer selbstorganisierter verteilter Systeme erarbeitet und anhand von beispielhaften Systemarchitekturen illustriert.				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Vertiefte Kenntnisse im Bereich Verteilte Systeme (z.B. Besuch einer VL Rechnernetze, Kommunikationsnetze o.ä.)				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektabgabe				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				

11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr.-Ing. Martin Hoffmann
12	Sonstige Informationen Literaturhinweise: Zeitschriften und Proceedings zum Thema Rolf P. Würtz (ed.): Organic Computing (Understanding Complex Systems), Springer Verlag Berlin, 2008, 356 p., hardcover ISBN 978-3540776567 Peer-to-Peer Systems and Applications; Ralf Steinmetz und Klaus Wehrle (Hrsg.); 629 Seiten, Springer, ISBN 3-540-29192-X, 2005. Peer-to-Peer-Netzwerke; Peter Mahlmann und Christian Schindelbauer; 293 Seiten, Springer, ISBN 978-3-540-33991-5, 2007.

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Verteilte und eingebettete Systeme					
Kenn-Nr. 1.25	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 4 SWS	Kontaktzeit 6 SWS /90 h	Selbst-studium 210 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden besitzen Kenntnisse zum Einsatz von eingebetteten Systemen in Netzwerken. Sie sind vertraut mit der Softwareentwicklung für eingebettete Systeme sowie für deren Kommunikation. Der Aufbau von Sensor-/Aktornetzen ist den Studierenden ebenso bekannt, wie der von Bussystemen für eingebettete Systeme mit Zeitbedingungen.				
3	Inhalte Grundlagen der Kommunikation in verteilten eingebetteten Systemen - Algorithmen für verteilte Systeme - Bussysteme und Protokolle, z.B. Feldbussysteme - Zeit-Synchronisation - Sichere Kommunikation - Architektur von verteilten eingebetteten Systemen und Sensor-/Aktornetzen - Anwendungsbeispiele				
4	Lehrformen Seminaristischer Unterricht, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Vertiefte Kenntnisse im Bereich Verteilte Systeme (z.B. Besuch einer VL Rechnernetze, Kommunikationsnetze o.ä.)				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektabgabe				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestehen der Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 10/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Matthias König, Prof. Dr. Martin Hoffmann				
12	Sonstige Informationen Literatur: • G.J.Pottie and W. J. Kaiser, Principles of Embedded Networked System Design, Cambridge University Press, 2005				

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Umweltinformatik

Kenn-Nr. 1.26	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS /90 h	Selbststudium 210 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Ziel ist die Förderung der Selbstständigkeit und praktischen Problemlösungskompetenz sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten in dem interdisziplinären Wissensgebiet der Umweltinformatik. Die Studenten erwerben Kompetenzen in der Verknüpfung von Wissen aus der Physik und der Informatik mit Elementen der psychologischen Verhaltensforschung. Sie engagieren sich mit Ihren Arbeiten für den Umweltschutz und lernen Ihre gesellschaftliche Verantwortung als Softwareentwickler und Informatiker im zukunftsfähigen positiven Sinne zu nutzen.</p> <p>Es findet eine Wissensvertiefung im Bereich der Umweltinformatik statt und es werden praktische Erfahrungen in der Projektplanung und Realisierung eines Forschungsprototyps in Form einer Software gesammelt.</p>				
3	<p>Inhalte</p> <p>Interdisziplinäre Ansätze sind typisch für forschungsrelevante Anwendungen der Informatik. In diesem Fach lehren zwei Dozenten aus dem Wissensgebiet der Dünnschicht-Physik und der Angewandten Informatik die grundlegenden Elemente zu den jeweiligen projektrelevanten Themen, welche aus aktuellen Themen aus dem Forschungsschwerpunkt „Soziale Mobilisierungsstrategien im Politikfeld Klimaschutz“ und dem Solar Computing Lab der Hochschule stammen sollen.</p> <p>Spezielle Methoden der Umweltinformatik sollen je nach Projektinhalten und Anwendungsfeld in der Veranstaltung vom Dozenten vermittelt werden, durch die Studierenden analysiert und in den projektspezifischen Implementierungen zum Einsatz kommen.</p> <p>Auswahl an Themen zur Wissensvermittlung:</p> <ul style="list-style-type: none"> • Regenerative Energien • Physikalische Grundzüge der Photovoltaik • Energie-Monitoring • Qualitätssicherung erneuerbaren Energieerzeugungssysteme • Energieeffiziente Gebäude • Politisch-wirtschaftliche Einflüsse auf die Umweltinformatik • Soziale und technische Aspekte von energieeffizientem Umweltverhalten • SW-Useability im Kontext der Verhaltensänderung • Maschinelles Lernen, Support Vector Maschins • Neuronale Informationsverarbeitung, probabilistisches Schliessen, Kohonen Karten (Anwendung von z.B. Matlab oder Java) • Bildverarbeitung und Mustererkennung (Anwendung von z.B. LabView und Programmieren in C++) <p>Ablauf der Projekte</p> <p>Im Fokus steht weitestgehend die selbstständige Bearbeitung einer komplexeren Aufgabenstellung im Rahmen eines Forschungs- und Entwicklungsprojekts im Team, welches auch in Kooperation mit Forschungs- und Entwicklungsabteilungen von Firmen bearbeitet werden kann. In der Regel bestehen die Projektgruppen aus 2-4 Studierenden, die sich frei zusammen finden, einen Projektleiter aus ihren Reihen wählen und nach mit den Dozenten vereinbarten Vorgehensmodellen entwickeln sollen. Die Dozenten definieren die interdisziplinäre Zielsetzung und führen einen regelmäßigen Diskurs über den Fortgang des Projekts. Sie vereinbaren außerdem mit dem Team Meilensteine sowie Kommunikations- und Kooperationsformen.</p>				
4	<p>Lehrformen</p> <p>Seminaristischer Unterricht, Praktikum</p>				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: -</p>				

	Inhaltlich: Programmieren in Java, Matlab oder C++. Grundlagen der Künstlichen Intelligenz, der Webentwicklung, Softwaretechnik, Datenbanken und der Mobilien Applikationsentwicklung
6	Prüfungsvoraussetzungen Keine
7	Prüfungsformen Projektarbeit
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) Keine
10	Stellenwert der Note für die Endnote 10/90
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Ing. Grit Behrens, Prof. Dr. rer. nat. Frank Hamelmann
12	Sonstige Informationen Tom Mitchell: "Machine Learning", McGraw Hill, 1997 , ISBN 0070428077 Volker Quaschnig: "Regenerative Energiesysteme", Hanser 2011, ISBN 9783446427327 Djamila Rekioua: "Optimization of Photovoltaic power Systems", Springer 2012, ISBN 9781447123484

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“ Musik informatik

Kenn-Nr. 1.27	Work-load 300 h	Credits 10 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage je- des Semester	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS /90 h	Selbststudium 210 h	geplante Gruppengröße Seminaristischer Unterricht 60 Praktikum 15	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Musikindustrie ist mit etwa 2 Milliarden Euro Umsatz allein in Deutschland eine der wichtigsten Zielindustrien für den Bereich der Medieninformatik. Die Musikinformatik befasst sich mit allen computerbasierten Techniken und der Entwicklung von Anwendungen zur Komposition, Produktion, Vertrieb, Abrechnung/Lizenzen und dem Genuss von Musik und anderen Audioprodukten. Darüber hinaus sind spezielle Aspekte des Musikmanagements, der Musikwirtschaft und der technischen Unterstützung kreativer Prozesse Musikschaffender Gegenstand des Fachgebiets. Die Studierenden erarbeiten sich anhand eines komplexen Projektes aus dem Bereich Musikinformatik einen wissenschaftlichen Zugang zu diesem wichtigen Teilgebiet der Medieninformatik.</p> <p>Das vom Dozenten gestellte Projekt dient zur Erarbeitung und der Anwendung wissenschaftlicher Methoden der Informatik insbesondere die erfindende Methode wissenschaftlicher Erkenntnisgewinnung, physikalischer und mathematischer Methoden zur Klangerzeugung, Komposition, Mastering etc. Die Studierenden beziehen dabei Erkenntnisse über die Musik als universelles kulturelles Phänomen in Ihre Überlegungen ein und machen sich dazu mit wissenschaftlicher Literatur aus Anthropologie, Psychologie und den Kulturwissenschaften vertraut. Die Studierenden sammeln praktische Erfahrungen in der Projektarbeit bei der wissenschaftlichen Recherche, beim Design und der Implementierung von Forschungsprototypen, der wissenschaftlichen Einordnung der Ergebnisse und der Präsentation und Darstellung der Ergebnisse in typischen Formaten der wissenschaftlichen Veröffentlichung</p>				
3	<p>Inhalte</p> <p>Die Musikinformatik verwendet Methoden und Erkenntnisse aus verschiedenen Bereichen der Informatik, Physik, Mathematik und den Kulturwissenschaften. Das Anwenden solcher Methoden unter kritisch-wissenschaftlicher Betrachtungsweise ist wesentlicher Inhalt der Veranstaltung.</p> <p>Auswahl an Themen:</p> <ul style="list-style-type: none"> • Mathematische Grundlagen der Musik • Physikalische Grundlagen der Musik • Analoge und digitale Klangerzeuger • Audiodigitalisierung und Audioformate • MIDI • Virtuelle Instrumente und VST • Digitale Klangbearbeitung und -veränderung • Spezielle Audio-Programmiersprachen • Audio-Bibliotheken für all-purpose Programmiersprachen, insb. C/C++ • Agogik und der menschliche Faktor • Die Musik als universell-menschliches Phänomen • Psychoakustik und Musikgenuss • Programmierung von DAWs <p>In Absprache mit dem Dozenten wählen die Studierenden Projektthemen aus und bearbeiten diese über ein Semester als Projektgruppe. Sie recherchieren den Stand der Technik und Wissenschaft, formulieren eine Forschungsfrage und erarbeiten das erforderliche Skillset. Sie bedienen sich aktueller Projektmanagementmethoden und -tools. Sie implementieren ausgewählte Teile der Modellierung in funktionierende Software. Sie ordnen die Ergebnisse in die aktuelle wissenschaftliche Diskussion ein und erstellen eine Darstellung der Ergebnisse in typischen Formaten der wissenschaftlichen Veröffentlichung.</p>				
4	Lehrformen Seminaristischer Unterricht / Praktikum				

5	Teilnahmevoraussetzungen Formal: - Inhaltlich:
6	Prüfungsvoraussetzungen Keine
7	Prüfungsformen Projektabgabe, mündliche Prüfung, Hausarbeit, Klausur
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) Keine
10	Stellenwert der Note für die Endnote 10/90
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Dominic Becking
12	Sonstige Informationen Literaturangaben: Aktuelle Zeitschriften und Proceedings zum Thema. Steppat, M.: Audioprogrammierung. Hanser, München, 2014. Boulanger, R., Lazzarini, V. (Hgg.): The Audio Programming Book. MIT Press, Cambridge USA, 2011. Mazzola, G.: Elemente der Musikinformatik. Birkhäuser, Basel, 2006. Loy, G.: Musimathics – the mathematical foundations of music, Vol. 1 u. 2. MIT Press, Cambridge USA, 2007. Gouveia, D.: Getting Started with C++ Audio Programming for Game Development. Packt Publishing, Birmingham, 2013. Brown, A. R.: Making Music with Java. o.O., 2005.

Wahlfach aus der Liste „Wissenschaftliche Anwendungen der Informatik“					
Domainspezifische Sprachen					
Kenn-Nr. 1.28	Work-load 300 h	Credits 10 cps	Studiensemester 1./2. Sem.	Häufigkeit des Angebots Bei Nachfrage alle 2 Semester	Dauer 1 Semester
1	Lehrveranstaltungen Vorlesungen 2 SWS Praktikum 4 SWS	Kontaktzeit 6 SWS /90 h	Selbststudium 210 h	geplante Gruppengröße Vorlesung 60 Praktikum 15	
2	Lernergebnisse (learning outcomes) / Kompetenzen Das Ziel ist die Förderung der Selbstständigkeit und praktischen Problemlösungskompetenz sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten. Das vom Dozenten gestellte Projekt dient der Wissensvertiefung im Bereich der Entwicklung von Domainspezifischen Sprachen und als praktische Erfahrung in der Projektplanung und -realisierung eines Forschungsprototyps.				
3	<p>Inhalte</p> <p>Domainspezifische Sprachen spielen eine immer größere Rolle in vielen Bereichen der Informatik, beispielsweise bei der Modellierung von Softwarearchitekturen (etwa AUTOSAR), beim Requirements Engineering, bei der Softwarequalitätssicherung oder einfach nur zur Unterstützung der Programmierer bei ihrer täglichen Arbeit. Ein wichtiger Aspekt ist dabei die Unterstützung des Anwenders einer DSL durch angepasste Werkzeuge.</p> <p>Die Vorlesung gibt einen Überblick über Domainspezifische Sprachen und führt in die Modellierung von DSL mit Hilfe von Xtext und Xtend ein. Eine Analyse der Methoden und deren Umsetzungen im Projektkontext führen die Studierenden durch.</p> <p>Auswahl der Themen:</p> <ul style="list-style-type: none"> • Überblick über Programmierparadigmen und -konzepte • Formale Sprachen, Syntax und Grammatik • Domainspezifische Sprachen (intern, extern) • Modellierung von Domänenwissen, Formalisierung als Sprache • Implementierung von DSL mit Xtext und Xtend <ul style="list-style-type: none"> ○ Einführung in die Xtext-Sprache ○ Einführung in die Xtend-Programmiersprache ○ Codegenerierung und Release der DSL und Tools ○ Qualitätssicherung (Debugging, Test und Validierung der DSL) ○ Xtext-Komponenten mit Dependency Injection anpassen <p>Ablauf der Projekte</p> <p>Weitgehend selbstständige Bearbeitung einer komplexeren Aufgabenstellung im Rahmen eines Forschungs- und Entwicklungsprojekts im Team, welches auch in Kooperation mit F&E-Abteilungen von Firmen bearbeitet werden kann. In der Regel bestehen die Projektgruppen aus 3-5 Studierenden, die sich frei zusammen finden und einen Projektleiter aus ihren Reihen wählen und nach mit dem Dozenten vereinbarten Vorgehensmodellen entwickeln. Der Dozent definiert die Zielsetzung und führt einen regelmäßigen Diskurs über den Fortgang des Projekts. Er vereinbart außerdem mit dem Team Meilensteine sowie Kommunikations- und Kooperationsformen.</p>				
4	Lehrformen Vorlesung, Praktikum				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Spezielle Methoden der Programmierung, Compilerbau				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Projektabgabe				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) Keine
10	Stellenwert der Note für die Endnote 10/90
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Carsten Gips
12	Sonstige Informationen Literatur: <ul style="list-style-type: none"> • Parr, T.: "Language Implementation Pattern", Pragmatic Programmers, 2010 • Voelter, M.: "DSL Engineering: Designing, Implementing and Using Domain-Specific Languages", CreateSpace Independent Publishing Platform, 2013 • Bettini, L.: "Implementing Domain-Specific Languages with Xtext and Xtend", PACKT Publishing, 2013 • Clausen, A.: "Programmiersprachen", Spektrum Akademischer Verlag, 2011 • Ghosh, D.: "DSLs in Action", Manning, 2011

Komplexitätstheorie					
Kenn-Nr. 2.1	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Übung 2 SWS	Kontaktzeit 4 SWS /60 h	Selbststudium 90 h	geplante Gruppengröße Seminaristischer Unterricht 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Im Rahmen dieser Veranstaltung werden den Studierenden die Grundbegriffe und Methoden der Komplexitätstheorie nahegebracht. Die Studierenden sind in der Lage, diese Methoden auf konkrete algorithmische Probleme anzuwenden. Sie lernen Probleme bzgl. Ihrer Komplexität einzuordnen und so geeignete algorithmische Techniken herauszufiltern. Sie erkennen die prinzipiellen Grenzen von Lösungsmöglichkeiten mit Rechnerhilfe.				
3	Inhalte Dieser Seminaristische Unterricht trifft eine Auswahl unter den Ergebnissen der Komplexitätstheorie, so dass die Bedeutung der Komplexitätstheorie für eine moderne Informatik in den Mittelpunkt rückt. Folgende Inhalte werden erlernt: <ul style="list-style-type: none"> • Komplexitätsklassen, P vs. NP • Reduktionen und Vollständigkeit • Platzkomplexität • Hierarchiesätze • Relativierung und Orakel-Turingmaschinen • Schaltkreiskomplexität • Polynomialzeit-Hierarchie • Probabilistische Komplexitätsklassen • Interaktive Beweissysteme 				
4	Lehrformen Seminaristischer Unterricht, Übung				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Algorithmen und Datenstrukturen, Grundlagen der theoretischen Informatik				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Mündliche Prüfung, Klausur				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christoph Thiel				
12	Sonstige Informationen Balcazar / Diaz / Gabarro, Structural Complexity I und II, Springer, 2011. C. H. Papadimitriou. Computational Complexity. Addison-Wesley. Reading. 1995. U. Schöning, Theoretische Informatik - kurz gefasst, Spektrum Akademischer Verlag; 2008 I. Wegener, Komplexitätstheorie: Grenzen der Effizienz von Algorithmen, Springer; Auflage: 2003				

Formale Modelle der Informatik					
Kenn-Nr. 2.2	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht Praktikum	2 SWS 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst-studium 90 h	geplante Gruppengröße Seminaristischer Unterricht Praktikum
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden dieses Moduls verfügen über ein detailliertes Wissen und Verständnis unterschiedlicher Spezifikations-sprachen. Die Studierenden sind in der Lage, Anforderungen formal zu präzisieren und Ansätze zur automatischen und manuellen Verifikation zu nutzen. Ebenso lernen die Studierende unterschiedliche formale Modelle, die bei der Entwicklung sequenzieller und verteilter Systeme eingesetzt werden können.</p> <p>Studierende, die an dieser Lehrveranstaltung teilgenommen haben, sollen</p> <ul style="list-style-type: none"> • die theoretischen Grundlagen formaler Modelle verstehen und die Methoden formaler Modelle auf konkrete Problemstellungen anwenden • anhand einiger ausgewählter Beispiele formale Modelle im Detail kennenlernen • die Grundbegriffe der formalen Modellierung, Spezifikation und Verifikation von objekt-orientierten Softwaresystemen beherrschen • den praktischen Umgang mit Spezifikations-sprachen für objektorientierte Software wie OCL und JML erlernen • das grundlegende methodische Vorgehen bei der Verifikation erlernen • selbständig mit Unterstützung von entsprechenden Verifikationswerkzeugen praktische Übungsaufgaben lösen können <p>Die erworbenen praktischen und theoretischen Fähigkeiten erweitern die formalen und mathematischen Kompetenzen in zukünftigen beruflichen Analyse-, Entwurfs- und Programmier-tätigkeiten.</p>				
3	<p>Inhalte</p> <p>1.1. Grundlagen, Motivation und Orientierung 1.2. Logik 1.2.1. Aussagenlogik 1.2.2. Prädikatenlogik 1.3. Verifikation 1.3.1. Vorgehensweise 1.3.2. Hoare Kalkül 1.4. Formale Spezifikation 1.4.1. Object Constraint Language (OCL) 1.4.2. Java Modelling Language (JML) 1.5. Ausblick und weiterführende Themen 1.5.1. Einführung in das Modelchecking 1.5.2. Spezifikation, Simulation und Verifikation von verteilten Systemen 1.5.3. Konkretisierung von Anforderungen mit Hilfe temporaler Logik 1.5.4. Erweiterung von Spezifikationen von verteilten Systemen um einen Zeitbegriff</p>				
4	<p>Lehrformen Seminaristischer Unterricht, Praktikum</p>				
5	<p>Teilnahmevoraussetzungen Formal: - Inhaltlich: Software Engineering, UML, Programmiermethoden, Objektorientierte Programmierung</p>				
6	<p>Prüfungsvoraussetzungen Keine</p>				
7	<p>Prüfungsformen Performanzprüfung</p>				
8	<p>Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Teilleistungen der Performanzprüfung</p>				
9	<p>Verwendung des Moduls (in anderen Studiengängen) Keine</p>				
10	<p>Stellenwert der Note für die Endnote 5/90</p>				

11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Jörg Brunsmann
12	Sonstige Informationen Das Modul wird in Englischer Sprache gehalten. Literatur: Beckert, S., Hahnle, R., Schmitt, P.H.: Verification of Object-Oriented Software: The KeY Approach, Springer Berlin Heidelberg, ISBN 978-3540689775 Kleuker, S.: Formale Modelle der Softwareentwicklung, Vieweg+Teubner, 2009 Warner J., Kleppe A.: The Object Constraint Language, Addison-Wesley Monin, J.-F.: Understanding Formal Methods, Springer

Zuverlässige und sichere Softwaresysteme					
Kenn-Nr. 2.3	Work-load 150 h	Credits 5 cps	Studien-semester 1./2. Sem.	Häufigkeit des Angebots Jedes WS	Dauer 1 Semester
1	Lehrveranstaltungen Seminaristischer Unterricht 2 SWS Übung 2 SWS	Kontaktzeit 4 SWS /60 h	Selbst- studium 90 h	geplante Gruppengröße Seminaristischer Unterricht 60 Übung 30	
2	Lernergebnisse (learning outcomes) / Kompetenzen Im Rahmen dieser Veranstaltung werden die Studierenden mit den Grundlagen und Modellen der Zuverlässigkeit und Sicherheit von Softwaresystemen vertraut gemacht. Sie kennen typische Standards zur Bewertung der Zuverlässigkeit und Sicherheit von Softwaresystemen und können diese exemplarisch anwenden. Weiterhin kennen sie Methoden und Standards des Software Engineering für die Spezifikation und den Entwurf zuverlässiger und sicherer Softwaresysteme und können diese anwenden.				
3	Inhalte <ul style="list-style-type: none"> • Grundlagen und Modelle für Zuverlässigkeit und Sicherheit; • Beschreibungstechniken für zuverlässige und sichere Softwaresysteme • Standards für die Bewertung und Entwicklung zuverlässiger und sicherer Softwaresysteme • Schwachstellen- und Risikobasierte Vorgehensmodelle zur Entwicklung zuverlässiger und sicherer Softwaresysteme; • Fallstudien in Forschung und Industrie 				
4	Lehrformen Seminaristischer Unterricht, Übung				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Grundlegende Begriffe der Sicherheit und Zuverlässigkeit, Software-Engineering, Betriebssysteme, Verteilte Systeme und Kommunikationsnetze, Datenbanken				
6	Prüfungsvoraussetzungen Keine				
7	Prüfungsformen Klausur, mündliche Prüfung				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Klausur				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Prof. Dr. Christoph Thiel				
12	Sonstige Informationen Literatur: N. Storey, Safety Critical Computer Systems, Addison Wesley, 1996 C. P. Pfleeger, Security in Computing, 4th ed., Prentice Hall 2007 J. Humble, D. Farley, Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation , Addison-Wesley 2010				

Masterarbeit					
Kenn-Nr. 3.1	Work-load 720 h	Credits 24 cps	Studien-semester 3. Sem.	Häufigkeit des Angebots Jedes Semester	Dauer 1 Semester
1	Lehrveranstaltungen 0,5 SWS Individuelle dozentengebundene Betreuung	Kontaktzeit 20 h	Selbststudium 700 h	geplante Gruppengröße 1	
2	Lernergebnisse (learning outcomes) / Kompetenzen Fähigkeit ein komplexes, praxisbezogenes Informatik-Thema selbstständig und mit wissenschaftlichen Methoden zu bearbeiten, d.h. Problemstellung analysieren, Lösungsmöglichkeiten aufzeigen, in den Stand der Wissenschaft/Technik einordnen, implementieren und abschließend bewerten. Fähigkeit zum Verfassen einer anspruchsvollen wissenschaftlichen Ausarbeitung zum Thema.				
3	Inhalte Mit der Masterarbeit soll unter Beweis gestellt werden, dass Studierende in der Lage sind, eine komplexe, umfangreiche und fachlich anspruchsvolle Fragestellung mit wissenschaftlichen Methoden innerhalb eines begrenzten Zeitraums zu lösen und das dabei erworbene theoretische und praktische Wissen nachvollziehbar auf hohem Niveau zu dokumentieren. <ol style="list-style-type: none"> 1. Konkretisieren der Aufgabenstellung 2. Erstellung eines Zeitplans 3. Evaluation und Aufstellung der zu verwendenden Techniken und Methoden 4. Erstellung eines Software-Konzeptes 5. Implementierung und Dokumentation der Software-Lösung 6. Gesamtbetrachtung, Test und Bewertung der Lösung 7. Darstellung der Lösung in Form der Master-Arbeit. Im Gegensatz zur Bachelorarbeit wird hier ein anspruchsvolleres und evtl. umfangreicheres Thema auf einem wissenschaftlich höheren Niveau über einen längeren Zeitraum bearbeitet.				
4	Lehrformen Individuelle dozentengebundene Betreuung Selbständige Erstellung der Masterarbeit				
5	Teilnahmevoraussetzungen Formal: - Inhaltlich:				
6	Prüfungsvoraussetzungen Fristgemäße Abgabe der Masterarbeit im Prüfungsamt (MPO §25)				
7	Prüfungsformen Von zwei Prüfern bewertete Masterarbeit				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Masterarbeit				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 24/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Alle Dozentinnen und Dozenten des Studienbereichs Informatik nach § 8 MPO: die mindestens die entsprechende Masterprüfung an einer Hochschule oder eine vergleichbare Prüfung abgelegt haben oder eine vergleichbare Qualifikation erworben haben und im Masterstudium eine einschlägige selbständige Lehrtätigkeit ausgeübt haben.				
12	Sonstige Informationen Die Literatur wird abhängig vom Thema der Arbeit von den Dozenten vorgeschlagen und dann von den Studierenden selbstständig aktualisiert, erweitert und ergänzt.				

Kolloquium					
Kenn-Nr. 3.2	Work-load 180 h	Credits 6 cps	Studien-semester 3. Sem.	Häufigkeit des Angebots Jedes Semester	Dauer 1 Semester
1	Lehrveranstaltungen	Kontaktzeit	Selbststudium 180 h	geplante Gruppengröße	
2	Lernergebnisse (learning outcomes) / Kompetenzen Der Studierende ist befähigt, die Ergebnisse der Masterarbeit, ihre fachlichen Grundlagen, ihre fachübergreifenden Zusammenhänge und ihre außerfachlichen Bezüge mündlich darzustellen, selbständig zu begründen und ihre Bedeutung für die Praxis einzuschätzen.				
3	Inhalte Kolloquiumsvortrag				
4	Lehrformen -				
5	Teilnahmevoraussetzungen Zugelassen wird, wer die in § 26 Abs. 3 MPO genannten Voraussetzungen für die Zulassung zur Masterarbeit nachgewiesen hat, alle bis auf zwei studienbegleitende Prüfungen bestanden hat und die Masterarbeit mindestens mit der Note 4,0 bewertet wurde.				
6	Prüfungsvoraussetzungen Masterarbeit im Prüfungsamt eingereicht				
7	Prüfungsformen Mündliche Prüfung, §26, Abs. 4 MPO				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) Keine				
10	Stellenwert der Note für die Endnote 6/90				
11	Modulbeauftragte/r und hauptamtlich Lehrende Alle Dozentinnen und Dozenten des Studienbereichs Informatik nach § 8 MPO: die mindestens die entsprechende Masterprüfung an einer Hochschule oder eine vergleichbare Prüfung abgelegt haben oder eine vergleichbare Qualifikation erworben haben und im Masterstudium eine einschlägige selbständige Lehrtätigkeit ausgeübt haben.				
12	Sonstige Informationen				