

## **AMTLICHE MITTEILUNGEN**

**VERKÜNDUNGSBLATT DER UNIVERSITÄT PADERBORN AM.UNI.PB**

**AUSGABE 44.17 VOM 16. JUNI 2017**

---

## **PRÜFUNGSORDNUNG**

**FÜR DEN MASTERSTUDIENGANG INFORMATIK DER  
FAKULTÄT FÜR ELEKTROTECHNIK, INFORMATIK UND MATHEMATIK  
AN DER UNIVERSITÄT PADERBORN**

**VOM 16. JUNI 2017**

**Prüfungsordnung für den Masterstudiengang Informatik  
der Fakultät für Elektrotechnik, Informatik und Mathematik an der Universität Paderborn**

**vom 16. Juni 2017**

Aufgrund des § 2 Absatz 4 und des § 64 Absatz 1 des Gesetzes über die Hochschulen des Landes Nordrhein-Westfalen (Hochschulgesetz – HG) vom 16. September 2014 (GV.NRW. S. 547), zuletzt geändert durch Gesetz vom 15. Dezember 2016 (GV. NRW. S. 1154), hat die Universität Paderborn die folgende Prüfungsordnung erlassen:

## Inhaltsverzeichnis

I. Allgemeines .....	4
§ 1 Zweck der Prüfungen, Ziel des Studiums .....	4
§ 2 Akademischer Grad .....	4
§ 3 Studienbeginn .....	4
§ 4 Zugangsvoraussetzungen .....	4
§ 5 Regelstudienzeit und Studienumfang .....	7
§ 6 Module .....	7
§ 7 Anerkennung von Leistungen .....	8
II. Prüfungsorganisation .....	9
§ 8 Prüfungsausschuss .....	9
§ 9 Prüfende und Beisitzende .....	10
III. Masterprüfung .....	10
§ 10 Ziel, Umfang und Art der Masterprüfung .....	10
§ 11 Zulassung zur Masterprüfung .....	12
§ 12 Anmeldung und Prüfungsfristen .....	13
§ 13 Abschluss eines Moduls .....	13
§ 14 Prüfungsleistungen in den Modulen .....	13
§ 15 Formen der Prüfungsleistungserbringung in den Modulen, Studienleistungen und qualifizierte Teilnahme .....	14
§ 16 Bewertung von Leistungen in den Modulen .....	15
§ 17 Abschlussarbeit .....	16
§ 18 Annahme und Bewertung der Masterarbeit .....	17
§ 19 Bewertung der Masterprüfung und Bildung der Gesamtnote .....	18
§ 20 Wiederholung von Prüfungsleistungen, Kompensation .....	18
§ 21 Zusatzmodule .....	19
§ 22 Versäumnis, Rücktritt, Täuschung, Ordnungsverstoß, Schutzvorschriften und Studierende mit Familienaufgaben .....	19
§ 23 Erfolgreicher Abschluss des Studiums, endgültiges Nichtbestehen .....	21
§ 24 Zeugnis, Transcript of Records, Diploma Supplement .....	22
§ 25 Masterurkunde .....	22
§ 26 Einsicht in die Prüfungsakten .....	22
IV. Schlussbestimmungen .....	23
§ 27 Ungültigkeit der Masterprüfung .....	23
§ 28 Aberkennung des Mastergrades .....	23
§ 29 Übergangsbestimmungen .....	24
§ 30 Inkrafttreten und Veröffentlichung .....	24
Anhang 1: Module und Prüfungsformen .....	25
Anhang 2: Nebenfachvereinbarungen für die Standardnebenfächer im Masterstudiengang Informatik .....	29
Anhang 3: Studienverlaufsplan .....	31
Anhang 4: Modulhandbuch .....	31

## I. Allgemeines

### § 1

#### Zweck der Prüfungen, Ziel des Studiums

- (1) Die Masterprüfung bildet einen zweiten berufsqualifizierenden Abschluss des Studiums der Informatik.
- (2) Durch die Masterprüfung soll festgestellt werden, ob die Studierenden die in einem vorangegangenen Bachelorstudiengang erworbenen für die Berufspraxis notwendigen Kenntnisse erweitert und in ausgewählten Bereichen vertieft haben, so dass sie die Fähigkeit besitzen, zur Problemlösung geeignete wissenschaftliche Methoden der Informatik anzuwenden und in ihrem Vertiefungsgebiet weiterzuentwickeln. Zu den Erfordernissen der Berufspraxis in der Informatik gehört auch die Fähigkeit, in fachlichen Angelegenheiten mündlich und schriftlich in englischer Sprache zu kommunizieren.
- (3) Das Studium vermittelt den Studierenden neben den allgemeinen Studienzielen des § 58 HG die Fähigkeit, in ihrer Arbeit die wissenschaftlichen Methoden der Informatik anzuwenden und weiter zu entwickeln und im Hinblick auf die Auswirkungen des technologischen Wandels verantwortlich zu handeln.

### § 2

#### Akademischer Grad

Ist das Masterstudium erfolgreich abgeschlossen, verleiht die Fakultät für Elektrotechnik, Informatik und Mathematik den akademischen Grad „Master of Science“, abgekürzt: „M.Sc.“.

### § 3

#### Studienbeginn

Der Studienbeginn ist das Wintersemester oder das Sommersemester.

### § 4

#### Zugangsvoraussetzungen

- (1) In den Masterstudiengang Informatik kann nur eingeschrieben werden, wer kumulativ
  1. das Zeugnis der Hochschulreife (allgemeine oder einschlägig fachgebundene) oder nach Maßgabe einer Rechtsverordnung das Zeugnis der Fachhochschulreife oder einen durch Rechtsvorschrift oder von der zuständigen staatlichen Stelle als gleichwertig anerkannten Vorbildungsnachweis oder die Voraussetzungen für in der beruflichen Bildung Qualifizierte besitzt oder die Voraussetzungen der Bildungsausländerhochschulzugangsverordnung erfüllt.
  2. einen Studienabschluss besitzt, der nachfolgende Voraussetzungen erfüllt:

- a) Es muss sich um einen ersten berufsqualifizierenden Hochschulabschluss mit einer Regelstudienzeit von mindestens sechs Semestern der Universität Paderborn oder einer staatlichen oder staatlich anerkannten Hochschule oder einer staatlichen oder staatlich anerkannten Berufsakademie handeln. Studienabschlüsse einer ausländischen staatlichen oder staatlich anerkannten Hochschule eröffnen den Zugang, sofern hinsichtlich der erworbenen Kompetenzen kein wesentlicher Unterschied zu einem Studienabschluss der Universität Paderborn nach Satz 1 besteht. Für ausländische Bildungsabschlüsse sind die von der Kultusministerkonferenz und der Hochschulrektorenkonferenz gebilligten Äquivalenzvereinbarungen oder entsprechende gesetzliche Regelungen zu berücksichtigen. Soweit Vereinbarungen und Abkommen der Bundesrepublik Deutschland mit anderen Staaten über die Gleichwertigkeit im Hochschulbereich (Äquivalenzabkommen) Studierende ausländischer Staaten abweichend von Satz 2 begünstigen, gehen die Regelungen der Äquivalenzabkommen vor. Im Übrigen soll bei Zweifeln über das Vorliegen oder Nichtvorliegen wesentlicher Unterschiede die Zentralstelle für ausländisches Bildungswesen gehört werden. Die Feststellung über die Voraussetzungen nach Satz 2 trifft der Prüfungsausschuss.
  - b) Der Studienabschluss muss mindestens 20 Leistungspunkte (LP) im Bereich Softwaretechnik, mindestens 20 LP im Bereich der Theoretischen Informatik, mindestens 15 LP im Bereich der Technischen Informatik, mindestens 15 LP im Bereich mathematischer Grundlagen der Informatik und mindestens 8 LP im Bereich Mensch-Maschine-Wechselwirkung beinhalten. Die Feststellung über die Voraussetzungen trifft der Prüfungsausschuss. Fehlen bis zu 30 Leistungspunkte, so kann die Einschreibung mit der Auflage erfolgen, die Anforderungen durch angemessene Studien im Rahmen von bis zu drei Auflagenkursen nachzuholen und durch das Bestehen zugehöriger Prüfungen nachzuweisen. Die Entscheidung hierüber trifft der Prüfungsausschuss. Die Prüfungen sollten innerhalb der ersten beiden Semester des Masterstudiengangs erbracht werden. Sie sind Voraussetzung für die Teilnahme an den Modulen der fachlich entsprechenden Focus Areas. Sie sind erneut bei Anmeldung zum Modul Master-Abschlussarbeit nachzuweisen. Näheres ist in Anhang 1 geregelt.
  - c) Der Studienabschluss muss mit einer Gesamtnote von mindestens 3,0 erfolgt sein.
3. ausreichende Sprachkenntnisse nach Maßgabe der Absätze 2 und 3 besitzt.
  4. als ausländische Studienbewerberin bzw. als ausländischer Studienbewerber, die bzw. der nicht durch oder aufgrund völkerrechtlicher Verträge Deutschen gleichgestellt ist, ihre bzw. seine Studierfähigkeit durch die Ergebnisse eines GRE Revised General Test nachweist. Erforderlich sind in der Regel mindestens 157 Punkte im Teil „Quantitative Reasoning“ und mindestens 4,0 Punkte im Teil „Analytical Writing“ des GRE Revised General Test. Bei einer sehr guten oder guten Abschlussnote des Abschlusses gemäß Nr. 2 ist der Nachweis des GRE Revised General Test nicht erforderlich. Studienbewerberinnen und Studienbewerber mit einer deutschen Hochschulzugangsberechtigung sind vom Nachweis der Studierfähigkeit ausgenommen.
- (2) Zum Masterstudiengang Informatik wird eingeschrieben, wer
1. englische Sprachkenntnisse besitzt, die nachgewiesen werden durch Zeugnisse oder andere Dokumente über
    - a) erfolgreich abgeschlossenen Schulunterricht in Englisch ab der Klasse 5 von mindestens fünf Jahren Dauer – als Bildungsinländer – oder
    - b) einen Sprachtest mindestens auf dem Niveau TOEFL 500 (paper and pencil) oder TOEFL 61 (internet-based) oder

- c) gleichwertige Kenntnisse (z. B. Cambridge First Certificate Note B oder IELTS mit dem Mindestergebnis 5.0)

2. als Studienbewerberin oder Studienbewerber, der ihre bzw. seine Zugangsvoraussetzungen nicht an einer deutschen Einrichtung erworben hat, ausreichende Kenntnisse der deutschen Sprache besitzt. Es bedarf eines Nachweises der sprachlichen Studierfähigkeit für die uneingeschränkte Zulassung oder Einschreibung zu allen Studiengängen. Näheres regelt die Ordnung für die Deutsche Sprachprüfung für den Hochschulzugang an der Universität Paderborn in der jeweils geltenden Fassung.

- (3) Alternativ zu Abs. 2 wird auch eingeschrieben, wer zwar nicht die dort geforderten Deutschkenntnisse besitzt, dafür aber über fundierte englische Sprachkenntnisse verfügt, die nachgewiesen werden durch Zeugnisse oder Dokumente über
- a) einen Bachelorabschluss im englischsprachigen Ausland<sup>1</sup> oder in einem als englischsprachig akkreditierten, inländischen Studiengang oder
  - b) Test of English as Foreign Language (TOEFL) "Internet-based" Test (iBT) mit einem Ergebnis von mindestens 80 Punkten oder
  - c) TOEFL "Paper-based" Test (PBT) mit einem Ergebnis von mindestens 550 Punkten oder
  - d) IELTS-Test mit einem Ergebnis von mindestens 6.5 oder
  - e) Cambridge Test – Certificate in Advanced English (CAE) Note B
- oder durch im Niveau gleichwertige Tests.
- (4) Nach Abs. 3 eingeschriebene Studierende müssen vor Abschluss ihres Masterstudiums Deutschkenntnisse der Kompetenzstufe A2 GER (Europäischer Referenzrahmen) nachweisen. Diese können in Deutschkursen erworben werden und mit maximal 12 LP im Rahmen des Studium Generale angerechnet werden.
- (5) Die Einschreibung ist abzulehnen, wenn
1. die in Absatz 1 bis 3 genannten Voraussetzungen nicht vorliegen,
  2. die Kandidatin bzw. der Kandidat eine nach der Prüfungsordnung erforderliche Prüfung in dem gewünschten Studiengang an einer Hochschule im Geltungsbereich des Grundgesetzes endgültig nicht bestanden hat oder
  3. die Kandidatin bzw. der Kandidat sonst eine nach der Prüfungsordnung erforderliche Prüfung in einem Studiengang an einer Hochschule im Geltungsbereich des Grundgesetzes endgültig nicht bestanden hat, wenn sowohl der erfolglose Studiengang eine erhebliche inhaltliche Nähe zum Masterstudiengang Informatik der Universität Paderborn als auch die endgültig nicht bestandene Prüfung eine erhebliche inhaltliche Nähe zu einer Prüfung eines Pflichtmoduls des Masterstudiengangs Informatik der Universität Paderborn aufweisen.

---

<sup>1</sup> Das sind im Rahmen dieser Ordnung Australien, Großbritannien, Irland, Kanada, Neuseeland und die Vereinigten Staaten von Amerika.

## § 5 Regelstudienzeit und Studienumfang

- (1) Die Regelstudienzeit beträgt einschließlich der Masterprüfung vier Semester. Es wird von einem Gesamtarbeitsaufwand (Workload) für die Studierenden entsprechend 120 LP (=3.600 Stunden) ausgegangen.
- (2) Das Masterstudium umfasst Module mit einem Gesamtumfang von 120 LP.
- (3) Alle Studierenden müssen Module und zugehörige Prüfungen im Hauptfach mit einem Umfang von mindestens 24 LP in englischer Sprache absolvieren.
- (4) Wird das Masterstudium Informatik vollständig in englischer Sprache studiert, muss mit einer geringen Einschränkung der Wahlfreiheit gerechnet werden. Das Gleiche gilt, wenn nur der in Abs. 3 geforderte Anteil an Modulen in englischer Sprache gewählt wird.
- (5) Die angegebenen LP entsprechen den im Rahmen des European Credit Transfer Systems (ECTS) zu vergebenden Punktzahlen. Ein LP entspricht einer Arbeitsbelastung von durchschnittlich 30 Stunden. Ein Semester umfasst in der Regel 30 LP und somit einen Arbeitsaufwand von 900 Stunden. Die Inhalte der Module sind so ausgewählt, dass dem durch die LP vorgesehenen Arbeitsaufwand Rechnung getragen wird.
- (6) Die Fakultät für Elektrotechnik, Informatik und Mathematik hat auf der Grundlage dieser Prüfungsordnung einen beispielhaften Studienverlaufsplan und ein Modulhandbuch erstellt. Diese Unterlagen geben insbesondere Aufschluss über die Ziele der einzelnen Module und über die den Modulen zugeordneten Lehrveranstaltungen sowie über die notwendigen Vorkenntnisse und Inhalte. Der beispielhafte Studienverlaufsplan und das Modulhandbuch liegen dieser Prüfungsordnung als Anlage bei. Das Modulhandbuch gibt den aktuellen Stand zum Zeitpunkt der Beschlussfassung über die Prüfungsordnung wieder. Das Modulhandbuch wird regelmäßig aktualisiert und auf den Internetseiten des Institutes für Informatik veröffentlicht.
- (7) Im Masterstudium ist für das Studium Generale und ein optionales Nebenfach zusammen ein Umfang von 12 LP vorgesehen. Im optionalen Nebenfach sind Prüfungen im Sinne von § 15 abzulegen. Im Studium Generale ist in einer der gewählten Veranstaltungen eine Prüfungsleistung zu erbringen. In den Veranstaltungen des Studium Generale, in denen keine Prüfungsleistung erbracht wird, ist ein Nachweis der qualifizierten Teilnahme zu erbringen.

## § 6 Module

- (1) Der Masterstudiengang Informatik wird in modularisierter Form angeboten. Module sind thematisch und zeitlich abgerundete, in sich abgeschlossene und mit LP versehene, prüfbare eigenständige Qualifikationseinheiten. Die Module haben einen Umfang von 5 bis 30 LP und können in der Regel innerhalb von ein bis zwei Semestern abgeschlossen werden.
- (2) Neben den Modulen Projektgruppe (20 LP) und Master-Abschlussarbeit (30 LP) sowie neben dem Studium Generale und Nebenfach (zusammen 12 LP) ist das Masterstudium in acht Wahlpflichtmodule (jeweils 6 LP) und zwei Seminare (je 5 LP) gegliedert.
- (3) Ein Modul kann Pflicht- und Wahlpflichtveranstaltungen enthalten. Enthält ein Modul Wahlpflichtveranstaltungen, werden diese aus einem Veranstaltungskatalog gewählt, der Teil der Modulbeschreibung ist.

## § 7 Anerkennung von Leistungen

- (1) Leistungen, die in anderen Studiengängen oder in Studiengängen an anderen staatlichen oder staatlich anerkannten Hochschulen, an staatlichen oder staatlich anerkannten Berufsakademien oder in Studiengängen an ausländischen staatlichen oder staatlich anerkannten Hochschulen erbracht worden sind, werden auf Antrag anerkannt, sofern hinsichtlich der erworbenen Kompetenzen kein wesentlicher Unterschied zu den Leistungen besteht, die ersetzt werden. Dabei ist kein schematischer Vergleich, sondern eine Gesamtbetrachtung im Hinblick auf den Anerkennungszweck der Fortsetzung des Studiums und des Ablegens von Prüfungen vorzunehmen. Für die Anerkennung von Leistungen in staatlich anerkannten Fernstudien oder in vom Land Nordrhein-Westfalen im Zusammenhang mit den anderen Ländern und dem Bund entwickelten Fernstudieneinheiten gelten Satz 1 und 2 entsprechend.
- (2) Für die Anerkennung von Leistungen an ausländischen Hochschulen sind die von der Kultusministerkonferenz und der Hochschulrektorenkonferenz gebilligten Äquivalenzvereinbarungen sowie Absprachen im Rahmen von Hochschulpartnerschaften zu beachten. Soweit Vereinbarungen und Abkommen der Bundesrepublik Deutschland mit anderen Staaten über Gleichwertigkeiten im Hochschulbereich (Äquivalenzabkommen) Studierende ausländischer Staaten abweichend von Absatz 1 begünstigen, gehen die Regelungen der Äquivalenzabkommen vor. Im Übrigen kann bei Zweifeln über das Vorliegen oder Nichtvorliegen wesentlicher Unterschiede die Zentralstelle für ausländisches Bildungswesen gehört werden.
- (3) Auf der Grundlage der Anerkennung nach Abs. 1 muss der Prüfungsausschuss auf Antrag der bzw. des Studierenden in ein Fachsemester einstufen.
- (4) Studienbewerberinnen und Studienbewerber, die aufgrund einer Einstufungsprüfung gemäß § 49 Abs. 12 HG berechtigt sind, das Studium aufzunehmen, werden die in der Einstufungsprüfung nachgewiesenen Kenntnisse und Fähigkeiten auf Leistungen anerkannt. Die Feststellungen im Zeugnis über die Einstufungsprüfung sind für den Prüfungsausschuss bindend.
- (5) Auf Antrag können vom Prüfungsausschuss sonstige Kenntnisse und Qualifikationen auf der Grundlage vorgelegter Unterlagen anerkannt werden, wenn diese Kenntnisse und Qualifikationen den Leistungen, die sie ersetzen sollen, nach Inhalt und Niveau gleichwertig sind.
- (6) Zuständig für die Anerkennungen nach den Absätzen 1 und 5 ist der Prüfungsausschuss. Vor Feststellungen über das Vorliegen oder Nichtvorliegen wesentlicher Unterschiede oder über die Gleichwertigkeit sind zuständige Fachvertreterinnen oder Fachvertreter zu hören. Wird die Anerkennung versagt, ist dies zu begründen.
- (7) Die antragstellende Person hat die für die Anerkennung erforderlichen Informationen (insbesondere die durch die Leistungen erworbenen Kenntnisse und Fähigkeiten und die Prüfungsergebnisse) in der vom Prüfungsausschuss festgelegten Form vorzulegen. Der Prüfungsausschuss hat über Anträge nach Absatz 1 spätestens innerhalb von zehn Wochen nach vollständiger Vorlage aller entscheidungserheblichen Informationen zu entscheiden.
- (8) Die Anerkennung wird im Zeugnis gekennzeichnet. Werden Leistungen anerkannt, sind die Noten, soweit die Bewertungssysteme vergleichbar sind, gegebenenfalls nach Umrechnung zu übernehmen und in die jeweilige Notenberechnung einzubeziehen. Ist keine Note vorhanden oder sind die Bewertungssysteme nicht vergleichbar, wird der Vermerk „bestanden“ aufgenommen.
- (9) Eine Leistung kann nur einmal anerkannt werden. Dies gilt auch für die Anerkennung sonstiger Kenntnisse und Qualifikationen.



## II. Prüfungsorganisation

### § 8 Prüfungsausschuss

- (1) Der Fakultätsrat der Fakultät für Elektrotechnik, Informatik und Mathematik bildet für den Masterstudiengang Informatik einen Prüfungsausschuss. Er ist insbesondere zuständig für
- die Organisation der Prüfungen und die Überwachung ihrer Durchführung,
  - die Einhaltung der Prüfungsordnung und die Beachtung der für die Durchführung der Prüfungen beschlossenen Verfahrensregelungen,
  - die Entscheidungen über Widersprüche gegen in Prüfungsverfahren getroffene Entscheidungen,
  - die Abfassung eines jährlichen Berichts an den Fakultätsrat über die Entwicklung der Prüfungen und Studienzeiten,
  - die weiteren durch diese Ordnung dem Prüfungsausschuss ausdrücklich zugewiesenen Aufgaben.

Der bzw. dem Vorsitzenden des Prüfungsausschusses sind bestimmte Aufgaben durch diese Ordnung zugewiesen. Darüber hinaus kann der Prüfungsausschuss die Erledigung von Angelegenheiten, die keine grundsätzliche Bedeutung haben, auf die Vorsitzende bzw. den Vorsitzenden übertragen; dies gilt nicht für Entscheidungen über Widersprüche und Berichte an den Fakultätsrat. Die bzw. der Vorsitzende berichtet dem Prüfungsausschuss über die von ihr bzw. ihm allein getroffenen Entscheidungen. Der Prüfungsausschuss und die bzw. der Vorsitzende des Prüfungsausschusses werden vom Zentralen Prüfungssekretariat unterstützt.

- (2) Der Prüfungsausschuss besteht aus der bzw. dem Vorsitzenden, der bzw. dem stellvertretenden Vorsitzenden und fünf weiteren Mitgliedern. Auf Vorschlag der jeweiligen Gruppe werden die bzw. der Vorsitzende, die bzw. der stellvertretende Vorsitzende und zwei weitere Mitglieder aus der Gruppe der Hochschullehrerinnen und Hochschullehrer, ein Mitglied aus der Gruppe der akademischen Mitarbeiterinnen und Mitarbeiter und zwei Mitglieder aus der Gruppe der Studierenden von ihren jeweiligen Vertreterinnen und Vertretern im Fakultätsrat gewählt. Entsprechend werden für die Mitglieder des Prüfungsausschusses mit Ausnahme der bzw. des Vorsitzenden und der bzw. des stellvertretenden Vorsitzenden Vertreterinnen und Vertreter gewählt. Die Amtszeit der Mitglieder aus der Gruppe der Hochschullehrerinnen und Hochschullehrer und des Mitgliedes der akademischen Mitarbeiterinnen und Mitarbeiter beträgt zwei Jahre und läuft vom 1. Oktober des Wahljahres bis zum 30. September des übernächsten Jahres. Die Amtszeit der Studierenden beträgt ein Jahr und läuft vom 1. Oktober des Wahljahres bis zum 30. September des nächsten Jahres. Eine Wiederwahl ist zulässig. Die Regelungen zur Geschlechtergerechtigkeit gemäß § 11c HG sind zu beachten.
- (3) Der Prüfungsausschuss ist Behörde im Sinne des Verwaltungsverfahrens- und des Verwaltungsprozessrechts.
- (4) Der Prüfungsausschuss ist beschlussfähig, wenn neben der bzw. dem Vorsitzenden oder der bzw. dem stellvertretenden Vorsitzenden und zwei weiteren Hochschullehrerinnen bzw. Hochschullehrern mindestens ein weiteres stimmberechtigtes Mitglied anwesend ist. Der Prüfungsausschuss beschließt mit einfacher Mehrheit. Bei Stimmgleichheit entscheidet die Stimme der bzw. des Vorsitzenden. Die studentischen Mitglieder des Prüfungsausschusses haben bei pädagogisch-wissenschaftlichen Entscheidungen, insbesondere bei der Beurteilung oder Anerkennung von Leistungen nur beratende Stimme.

- (5) Der Prüfungsausschuss wird von der bzw. dem Vorsitzenden einberufen. Die Einberufung muss erfolgen, wenn mindestens drei Mitglieder dies verlangen.
- (6) Die Sitzungen des Prüfungsausschusses sind nicht öffentlich. Die Mitglieder des Prüfungsausschusses, ihre Stellvertreterinnen und Stellvertreter, die Prüfenden und die Beisitzenden unterliegen der Amtsverschwiegenheit. Sofern sie nicht im öffentlichen Dienst stehen, sind sie durch die Vorsitzende bzw. den Vorsitzenden des Prüfungsausschusses zur Verschwiegenheit zu verpflichten.
- (7) Die Mitglieder des Prüfungsausschusses haben das Recht, der Abnahme der Prüfungen beizuwohnen.

## § 9

### Prüfende und Beisitzende

- (1) Die bzw. der Vorsitzende des Prüfungsausschusses bestellt die Prüfenden und Beisitzenden. Prüfende sind alle selbstständig Lehrenden der Veranstaltungen, in denen nach Maßgabe des Curriculums und der Modulbeschreibungen Prüfungsleistungen erbracht werden können. Als Beisitzerin bzw. Beisitzer kann nur bestellt werden, wer mindestens die entsprechende Masterprüfung oder eine vergleichbare Prüfung abgelegt hat.
- (2) Die Prüfenden sind in ihrer Prüfungstätigkeit unabhängig.
- (3) Die Kandidatin bzw. der Kandidat kann für die Masterarbeit und – wenn mehrere Prüfende zur Auswahl stehen – für die mündlichen Prüfungen Prüfende vorschlagen. Die Vorschläge sollen nach Möglichkeit Berücksichtigung finden. Ein Rechtsanspruch besteht nicht.
- (4) Der Prüfungsausschuss sorgt dafür, dass der Kandidatin bzw. dem Kandidat die Namen der Prüfenden rechtzeitig, in der Regel vier, mindestens aber zwei Wochen vor dem Termin der jeweiligen Prüfung, bekannt gegeben werden. Die Bekanntgabe im Campus Management System der Universität Paderborn ist ausreichend.

## III. Masterprüfung

### § 10

#### Ziel, Umfang und Art der Masterprüfung

- (1) Die Masterprüfung erstreckt sich auf die folgenden sogenannten Focus Areas
  1. Software Engineering
  2. Algorithm Design
  3. Networks and Communication
  4. Computer Systems
  5. Intelligence and DataEine dieser fünf Focus Areas ist als Specialization Area zu wählen.
- (2) Als Standardnebenfächer können gewählt werden:
  1. Elektrotechnik

2. Mathematik
3. Medienwissenschaft
4. Philosophie
5. Psychologie
6. Wirtschaftsinformatik
7. Wirtschaftswissenschaften

Für diese Nebenfächer existiert jeweils eine Nebenfachvereinbarung mit einem abgestimmten Modulangebot; die Nebenfachvereinbarungen sind im Anhang dieser Prüfungsordnung enthalten. Die Stundenpläne werden im Zuge der Erstellung des Vorlesungsverzeichnisses koordiniert.

Ein ausreichendes, vollständig englischsprachiges Angebot wird in den Nebenfächern Elektrotechnik und Wirtschaftsinformatik gewährleistet.

Auf Antrag kann der Prüfungsausschuss im Einzelfall ein anderes Fach als Nebenfach zulassen. In diesem Fall bestimmt er die zu erbringenden Prüfungsleistungen und teilt diese der Antragstellerin oder dem Antragsteller mit.

(3) Die Masterprüfung besteht aus

1. drei studienbegleitenden Modulprüfungen in Wahlpflichtmodulen mit einem Umfang von je 6 LP in der Specialization Area,
2. einer studienbegleitenden Modulprüfung in einem Wahlpflichtmodul mit einem Umfang von 6 LP in einer von der Specialization Area verschiedenen Focus Area,
3. vier studienbegleitenden Modulprüfungen in Wahlpflichtmodulen mit einem Umfang von je 6 LP aus beliebig wählbaren Focus Areas,
4. dem Modul Projektgruppe (20 LP),
5. zwei Seminar-Modulen (je 5 LP),
6. dem Modul Master-Abschlussarbeit (30 LP) einschließlich einer Arbeitsplanung (5 LP), eines Vortrages von etwa 45 Minuten Dauer,
7. Prüfungen in einem optionalen Nebenfach oder einer Prüfung im Rahmen des Moduls „Studium Generale“ mit einem Umfang von insgesamt mindestens 12 LP.

(4) Module und zugehörige Prüfungen im Hauptfach mit einem Umfang von mindestens 24 LP müssen in englischer Sprache absolviert werden. Dies bedeutet im Rahmen dieser Ordnung, dass Vorlesungen und Prüfungen in englischer Sprache gehalten werden.

(5) Die Bescheinigung nach § 2 Satz 2 über den Abschluss „Englischsprachiger Masterstudiengang Informatik“ wird ausgestellt, wenn

1. die Prüfung nach Abs. 3 Nr. 6 (Modul Master-Abschlussarbeit) vollständig in englischer Sprache absolviert worden ist und
2. solche nach Abs. 3 Nr. 1 bis 5 (Wahlpflichtmodule), mit Ausnahme von Modulen und Prüfungen im Umfang von höchstens 18 LP, gemäß der in Abs. 4 beschriebenen Form absolviert worden sind.

- (6) Im optionalen Nebenfach sind Modulprüfungen im Umfang von 12 LP abzulegen. Alternativ ist ein Studium Generale im Umfang von 12 LP zu absolvieren. In diesem Fall ist im Studium Generale in einer der gewählten Veranstaltungen eine Prüfungsleistung zu erbringen. In den Veranstaltungen des Studium Generale, in denen keine Prüfungsleistung erbracht wird, ist ein Nachweis der qualifizierten Teilnahme zu erbringen. Näheres ist für Standardnebenfächer dem Anhang 2 zu entnehmen. Bei Zulassung nach § 4 Abs. 3 können die geforderten Deutschkurse als Prüfungen im Studium Generale gemäß Abs. 3 Nr. 5 anerkannt werden. Im Studium Generale dürfen keine Informatikveranstaltungen abgelegt werden. Über Ausnahmen entscheidet der Prüfungsausschuss.

## § 11

### Zulassung zur Masterprüfung

- (1) Zu Prüfungen im Masterstudiengang Informatik kann nur zugelassen werden, wer an der Universität Paderborn für den Masterstudiengang Informatik eingeschrieben oder gemäß § 52 HG als Zweithörerin oder Zweithörer zugelassen ist. Auch während der Prüfungen müssen diese Erfordernisse gegeben sein.
- (2) Nach Maßgabe verfügbarer Kapazitäten und auf Antrag beim Prüfungsausschuss können über Absatz 1 hinaus Studierende des Bachelorstudiengangs Informatik, die in ihrem Bachelorstudiengang mindestens 152 abschlussrelevante LP erworben und die Bachelorarbeit angemeldet haben für ein Semester zu Modulen des Masterstudiengangs im Umfang von maximal 30 LP zugelassen werden. Davon ausgenommen ist das Modul Projektgruppe. Von der Regelung kann nur einmalig Gebrauch gemacht werden. Eine Wiederholung einer nicht bestandenen vorgezogenen Masterprüfung ist erst nach der Einschreibung in den Masterstudiengang Informatik möglich. Studierende haben keinen Anspruch darauf, zu einem späteren Zeitpunkt Zugang zum Masterstudiengang Informatik zu erhalten.
- (3) Zum Modul Master-Abschlussarbeit kann nur zugelassen werden, wer Module im Umfang von 48 LP (davon mindestens 18 LP in der gewählten Specialization Area) erfolgreich abgeschlossen hat. Im Fall der Einschreibung mit Auflagen gemäß § 4 muss zudem das Bestehen der zugehörigen Prüfungen nachgewiesen wurde. Die Ausgabe des Themas der Masterarbeit durch die Vorsitzende oder den Vorsitzenden des Prüfungsausschusses erfolgt unverzüglich nach der schriftlich vermerkten Annahme des Arbeitsplans durch die Betreuerin oder den Betreuer. Der Zeitpunkt der Ausgabe ist beim Zentralen Prüfungssekretariat aktenkundig zu machen.
- (4) Die Meldung zur Masterarbeit ist schriftlich über das Zentrale Prüfungssekretariat an die Vorsitzende bzw. den Vorsitzenden des Prüfungsausschusses zu stellen. Bei der Meldung zur Masterarbeit ist die Sprache, in der die Masterarbeit verfasst wird, anzugeben.
- (5) Die Zulassung zu einer Prüfung ist abzulehnen, wenn die in Abs. 1 und 3 genannten Voraussetzungen nicht erfüllt sind.
- (6) Als Nebenfach ist in der Regel ein beim ersten Abschluss studiertes Fach zu wählen; über Ausnahmen entscheidet der Prüfungsausschuss. Die Festlegung des Nebenfachs geschieht mit der ausdrücklichen Anmeldung als Nebenfachprüfung und dem Ablegen der ersten Prüfung in diesem Fach. Das Nichterscheinen oder der Rücktritt ohne triftige Gründe gem. § 22 steht dem Ablegen der Prüfung gleich. Die Kandidatin oder der Kandidat meldet ihre oder seine Teilnahme an einer Prüfung im Nebenfach jeweils spätestens zwei Wochen vor dem Prüfungstermin beim Prüfungsausschuss an.

## § 12 Anmeldung und Prüfungsfristen

- (1) Zu jedem Modul ist eine gesonderte Anmeldung über das Campus Management System erforderlich. Zudem ist zu jeder Prüfung eine gesonderte Meldung über das Campus Management System innerhalb der festgelegten Fristen erforderlich. Die Fristen der Prüfungsanmeldephasen werden auf den jeweiligen Informationsseiten des Campus Management Systems bekannt gegeben.
- (2) Die Anmeldung zu mündlichen Prüfungen gemäß § 13 Abs. 2, für die von der oder dem Lehrenden ein Prüfungsblock festgelegt wird, währenddessen die einzelnen Prüfungen stattfinden (Blockprüfungen), erfolgt innerhalb der festgelegten Fristen über das Campus Management System. Die Anmeldung zu Prüfungen gemäß § 13 Abs. 2, die als mündliche Prüfungen ohne Festlegung eines Prüfungsblocks angeboten werden (Individualprüfungen), erfolgt innerhalb der festgelegten Fristen über das Campus Management System. Der konkrete Prüfungstermin wird dabei von der Prüferin bzw. von dem Prüfer vergeben. Die Anmeldetermine für die Projektgruppe und die Seminare werden vor der Prüfungsanmeldephase von der bzw. dem Lehrenden festgelegt.
- (3) Die Prüfungen können abgelegt werden, sobald die für die Zulassung erforderlichen Voraussetzungen aus § 11 erfüllt sind, die Kandidatin bzw. der Kandidat sich gemäß der Absätze 1 und 2 angemeldet hat und zugelassen wurde.
- (4) Bei Modulen des gewählten Nebenfachs kommen bei Anmeldung, Abmeldung und Rücktritt die Regelungen der jeweils einschlägigen Prüfungsordnung des anbietenden Fachs zur Anwendung. Bei Modulen im Studium Generale kommen bei Anmeldung, Abmeldung, Rücktritt, Täuschung, Ordnungsverstoß und Bewertung der Prüfungsleistungen die Regelungen dieser Prüfungsordnung zur Anwendung.

## § 13 Abschluss eines Moduls

- (1) Jedes Modul wird durch eine Modulprüfung abgeschlossen. Die Modulprüfung findet im zeitlichen Zusammenhang mit dem Modul statt. Eine Modulprüfung besteht in der Regel aus einer Prüfung am Ende des Moduls (Modulabschlussprüfung). Die Modulprüfung kann aber auch aus mehreren Teilprüfungen (Modulteilprüfungen) bestehen. Besteht ein Modul aus mehreren Teilprüfungen, muss jede Teilprüfung bestanden sein. Die Modulnote entspricht der in der Modulprüfung erreichten Note.
- (2) Leistungspunkte können nur erworben werden, wenn das Modul erfolgreich abgeschlossen worden ist. Ein Modul ist erfolgreich abgeschlossen, wenn die Modulabschlussprüfung bzw. alle Modulteilprüfungen mit mindestens „ausreichend“ bewertet und die etwaig vorgesehenen qualifizierten Teilnahmen nachgewiesen wurden.

## § 14 Prüfungsleistungen in den Modulen

- (1) In den Modulen werden Prüfungsleistungen nach Maßgabe der Modulbeschreibungen erbracht. Die Noten der Modulprüfungen gehen in die Abschlussnote der Masterprüfung ein.

- (2) Sofern in den Modulbeschreibungen Rahmenvorgaben zu Form und/oder Dauer/Umfang von Prüfungsleistungen enthalten sind, setzt der Prüfungsausschuss im Benehmen mit der bzw. dem Prüfenden fest, wie die Prüfungsleistung konkret zu erbringen ist. In allen Modulen wird spätestens in der dritten Woche nach Vorlesungsbeginn von den jeweiligen Lehrenden bekannt gegeben, wie die Prüfungsleistungen erbracht werden können. Dies gilt entsprechend für den Nachweis der qualifizierten Teilnahme und Studienleistungen. Die Prüfungsleistungen beziehen sich jeweils auf die Inhalte und Kompetenzen der zugehörigen Lehrveranstaltungen.
- (3) Alle Prüfungen werden studienbegleitend abgelegt. Die Prüfungen finden in der Regel zweimal im Studienjahr statt. Die Wiederholung findet in der Regel im gleichen Semester, spätestens nach sechs Monaten, statt.

## § 15

### Formen der Prüfungsleistungserbringung in den Modulen, Studienleistungen und qualifizierte Teilnahme

- (1) Prüfungsleistungen können in Form von mündlichen Prüfungen oder in anderen Formen erbracht werden.

Die Bewertung ist den Studierenden außer bei mündlichen Prüfungen in der Regel spätestens sechs Wochen nach Leistungserbringung im Campus Management System der Universität Paderborn bekannt zu geben.

#### 1. Mündliche Prüfungen:

- In den mündlichen Prüfungen soll die Kandidatin bzw. der Kandidat nachweisen, dass sie bzw. er die Zusammenhänge des Prüfungsgebietes erkennt und spezielle Fragestellungen in diese Zusammenhänge einzuordnen vermag. Durch die mündlichen Prüfungen soll ferner festgestellt werden, ob die Kandidatin bzw. der Kandidat über ein breites Wissen um wissenschaftliche Methoden der Informatik verfügt.
- Mündliche Prüfungen werden vor zwei Prüfenden oder vor einer bzw. einem Prüfenden in Gegenwart einer bzw. eines sachkundigen Beisitzenden gemäß § 9 abgelegt. § 20 Absatz 5 bleibt unberührt. Vor der Festsetzung der Note beraten die Prüfenden bzw. hört die bzw. der Prüfende die Beisitzende bzw. den Beisitzenden in Abwesenheit der Kandidatin bzw. des Kandidaten.
- Mündliche Prüfungen dauern in der Regel mindestens 25 Minuten und höchstens 50 Minuten. Bei Gruppenprüfungen kann die Zeit angemessen verlängert werden.
- Die wesentlichen Gegenstände und Ergebnisse der mündlichen Prüfung sind in einem Protokoll festzuhalten. Das Ergebnis der Prüfung ist der Kandidatin bzw. dem Kandidaten im Anschluss an die mündliche Prüfung durch den oder die Prüfenden bekanntzugeben.
- Studierende, die sich in einem späteren Prüfungstermin der gleichen mündlichen Prüfung unterziehen wollen, werden nach Maßgabe der räumlichen Verhältnisse als Zuhörende zugelassen, sofern nicht eine Kandidatin oder ein Kandidat widerspricht. Die Zulassung erstreckt sich nicht auf die Beratung und Bekanntgabe des Prüfungsergebnisses.

#### 2. Prüfungsleistung im Seminar:

Im Seminar wird ein Seminarvortrag von 45 bis 60 Minuten Dauer gehalten und eine schriftliche Ausarbeitung angefertigt.

#### 3. Prüfungsleistung im Modul Projektgruppe

Im Modul Projektgruppe ist die erfolgreiche Bearbeitung von Projekten durch die Abgabe von Software und Dokumentation als phasenbezogene Prüfung nachzuweisen. Es wird eine Note für die Gesamtheit der bearbeiteten Projekte vergeben.

- (2) Die qualifizierte Teilnahme wird in der Regel durch eine Praktikumsarbeit mit anschließendem Gespräch nachgewiesen. Eine qualifizierte Teilnahme liegt dann vor, wenn die erbrachten Leistungen erkennen lassen, dass eine mehr als nur oberflächliche Beschäftigung mit den Gegenständen, die einer Aufgabenstellung zugrunde lagen, stattgefunden hat.
- (3) Als Studienleistung können Übungsaufgaben verlangt werden, die in der Regel wöchentlich als Hausaufgaben und/oder Präsenzaufgaben gestellt werden.
- (5) Im Studium Generale werden mündliche oder schriftliche Prüfungsleistungen erbracht. Dabei handelt es sich in der Regel um eine Klausur (maximal vier Stunden), eine Hausarbeit (maximal 25 Seiten) oder eine mündliche Prüfung (maximal 45 Minuten).

## § 16

### Bewertung von Leistungen in den Modulen

- (1) Die Noten für die einzelnen Prüfungsleistungen werden von den jeweiligen Prüfenden festgelegt. Für die Bewertung der Prüfungsleistungen sind folgende Noten zu verwenden:
 

1 = sehr gut:	eine ausgezeichnete Leistung
2 = gut:	eine Leistung, die erheblich über den durchschnittlichen Anforderungen liegt
3 = befriedigend:	eine Leistung, die durchschnittlichen Anforderungen genügt
4 = ausreichend:	eine Leistung, die trotz ihrer Mängel noch den Anforderungen genügt
5 = mangelhaft:	eine Leistung, die wegen erheblicher Mängel den Anforderungen nicht genügt
- (2) Zur differenzierten Bewertung können Zwischenwerte durch Absenken oder Anheben der einzelnen Noten um 0,3 gebildet werden. Dabei sind die Zwischennoten 0,7; 4,3; 4,7 und 5,3 ausgeschlossen.
- (3) Wird eine Prüfungsleistung von mehreren Prüfenden bewertet, wird die Note aus dem arithmetischen Mittel der Einzelnoten gebildet. Im Übrigen gelten Absatz 4 Satz 2 und 3 entsprechend.
- (4) Setzt sich eine Modulnote aus mehreren Noten zusammen, ist das arithmetische Mittel zu bilden. Abweichungen hiervon sind in den Modulbeschreibungen geregelt. Das Ergebnis ist nach der ersten Dezimalstelle hinter dem Komma abzuschneiden. Die Note lautet:
  - bei einem Durchschnitt bis einschließlich 1,5 = sehr gut,
  - bei einem Durchschnitt über 1,5 bis einschließlich 2,5 = gut,
  - bei einem Durchschnitt über 2,5 bis einschließlich 3,5 = befriedigend,
  - bei einem Durchschnitt über 3,5 bis einschließlich 4,0 = ausreichend,
  - bei einem Durchschnitt über 4,0 bis 5,0 = mangelhaft.

- (5) Zusätzlich zu den Prüfungsleistungen können freiwillig Leistungen im Rahmen eines Bonussystems erbracht werden (Bonusleistungen), die bewertet werden und die Modulnote nach einem festgelegten Schlüssel verbessern können. Die Bonusleistungen werden studienbegleitend und ausschließlich im Zusammenhang mit einer konkreten Veranstaltung erbracht. Als Erbringungsformen sind Hausaufgaben, Kurzvorträge oder Projektarbeit zulässig. Die Bonusleistungen sollen die Studierenden schrittweise auf nachfolgende Prüfungsleistungen vorbereiten. Ob in einer Veranstaltung Bonusleistungen erbracht werden können und der etwaige Schlüssel zur Verbesserung der Modulnote wird vom Prüfungsausschuss im Benehmen mit den jeweiligen Lehrenden festgelegt und spätestens in der dritten Woche nach Vorlesungsbeginn von den jeweiligen Lehrenden bekannt gegeben. Die Modulprüfung muss unabhängig von den Bonusleistungen bestanden werden. Die Bonusleistungen können die Modulnote um insgesamt maximal 0,7 verbessern.
- (6) Studienleistungen werden mit „bestanden“ oder „nicht bestanden“ bewertet.
- (7) Qualifizierte Teilnahmen sind nachzuweisen.

## § 17 Abschlussarbeit

- (1) Das Modul Master-Abschlussarbeit (30 LP) besteht aus der Arbeitsplanung (5 LP) und der Masterarbeit einschließlich eines Vortrags (25 LP). Die Thematik muss aus der Specialization Area gewählt werden.
- (2) Die Masterarbeit ist eine Prüfungsarbeit, die die wissenschaftliche Ausbildung abschließt und zeigen soll, dass die Kandidatin oder der Kandidat die Fähigkeit besitzt, innerhalb einer bestimmten Frist ein Problem der Informatik nach wissenschaftlichen Methoden zu bearbeiten. Die Aufgabenstellung soll so gestaltet werden, dass sie einem Arbeitsaufwand von fünf Monaten Vollzeitarbeit entspricht. Die Arbeit muss fünf Monate nach der Ausgabe abgegeben werden. Die Arbeit soll einen Umfang von in der Regel nicht mehr als 120 DIN A4-Seiten haben
- (3) Die Masterarbeit wird von einer von der bzw. dem Vorsitzenden des Prüfungsausschusses bestellten Person mit Prüferqualifikation nach § 9 gestellt und betreut werden. Auf Antrag der Kandidatin bzw. des Kandidaten kann der Prüfungsausschuss auch Prüfungsberechtigte zur Betreuung der Masterarbeit zulassen, die das von der Kandidatin bzw. dem Kandidaten gewählte Nebenfach vertreten. In diesem Fall benennt der Prüfungsausschuss eine zweite Prüferin bzw. einen zweiten Prüfer aus der Informatik, mit der bzw. mit dem der Arbeitsplan abgestimmt werden muss. Der Kandidatin bzw. dem Kandidaten ist Gelegenheit zu geben, Vorschläge für das Thema der Masterarbeit zu machen; dies begründet jedoch keinen Rechtsanspruch.
- (4) Auf Antrag sorgt die bzw. der Vorsitzende des Prüfungsausschusses dafür, dass die Kandidatin bzw. der Kandidat rechtzeitig ein Thema für eine Masterarbeit erhält. Der Zeitpunkt der Vergabe ist beim Zentralen Prüfungssekretariat aktenkundig zu machen.
- (5) Die Masterarbeit kann auch in Form einer Gruppenarbeit zugelassen werden, wenn der als Prüfungsleistung zu bewertende Beitrag der einzelnen Kandidatin bzw. des einzelnen Kandidaten aufgrund der Angabe von Abschnitten, Seitenzahlen oder anderen, objektiven Kriterien, die eine eindeutige Abgrenzung ermöglichen, deutlich unterscheidbar und bewertbar ist und die Anforderungen nach Abs. 2 erfüllt. Die Erstellung einer Gruppenarbeit mit Studierenden des Masterteilzeitstudiengangs ist nicht zulässig.



- (6) Das Thema und die Aufgabenstellung der Masterarbeit wird der Kandidatin bzw. dem Kandidaten schriftlich mitgeteilt. Das Thema kann nur einmal und nur innerhalb des ersten Monats nach der Ausgabe zurückgegeben werden. Wird das Thema der Masterarbeit nach dieser Frist zurückgegeben, gilt die Masterarbeit als nicht bestanden. Die Bearbeitungszeit beginnt mit der Vergabe des neuen Themas nach Annahme des neuen Arbeitsplans erneut. Ausnahmsweise kann der Prüfungsausschuss im Einzelfall auf begründeten Antrag, der spätestens eine Woche vor Ablauf der Abgabefrist beim Prüfungsausschuss gestellt werden muss, die Bearbeitungszeit um bis zu sechs Wochen verlängern, wenn die Gründe hierfür mit dem Thema der Arbeit zusammenhängen und die bzw. der Betreuende nach Abs. 3 dies befürwortet.
- (7) Bei Erkrankung innerhalb der Bearbeitungszeit kann auf Antrag der Kandidatin bzw. des Kandidaten die Frist für die Abgabe der Masterarbeit um höchstens vier Wochen verlängert werden. Dazu ist die unverzügliche Vorlage eines ärztlichen Attestes erforderlich. Es reicht eine ärztliche Bescheinigung über das Bestehen der Prüfungsunfähigkeit aus. Bestehen zureichende tatsächliche Anhaltspunkte, die eine Prüfungsfähigkeit als wahrscheinlich annehmen oder einen anderen Nachweis als sachgerecht erscheinen lassen, kann eine ärztliche Bescheinigung einer Vertrauensärztin oder eines Vertrauensarztes der Universität Paderborn auf Kosten der Universität verlangt werden. Gibt der Prüfungsausschuss dem Antrag statt, wird dies der Kandidatin bzw. dem Kandidaten schriftlich mitgeteilt. Die Verlängerung entspricht der Krankheitszeit; sie zieht keine Verlängerung der Regelstudienzeit nach sich. Überschreitet die Dauer der Erkrankung vier Wochen, so kann die Kandidatin bzw. der Kandidat nach Wahl die Arbeit innerhalb der um vier Wochen verlängerten Frist beenden oder ein neues Thema beantragen. Lehnt der Prüfungsausschuss den Antrag ab, wird dies der Kandidatin bzw. dem Kandidaten ebenfalls schriftlich mitgeteilt.
- (8) Der Arbeitsplan muss die folgenden Elemente enthalten: Beschreibung der zu bearbeitenden Aufgabe, Motivation der Arbeit, explizite Formulierung der Zielsetzung, Beschreibung der durchzuführenden Arbeiten, um das Ziel zu erreichen, einschließlich eines zugehörigen Zeitplans sowie eine Aufstellung einer vorläufigen Gliederung der schriftlichen Ausarbeitung.
- (9) Die Masterarbeit darf nicht, auch nicht auszugsweise, für eine abgeschlossene Prüfung angefertigt worden sein.
- (10) Bei der Abgabe der Masterarbeit hat die Kandidatin bzw. der Kandidat schriftlich zu versichern, dass sie bzw. er die Arbeit – bei einer Gruppenarbeit den entsprechend gekennzeichneten Anteil der Arbeit – selbständig verfasst und keine anderen als die angegebenen Quellen als Hilfsmittel benutzt sowie Zitate kenntlich gemacht hat. Der Arbeitsplan ist mit der Arbeit einzureichen.

## § 18

### Annahme und Bewertung der Masterarbeit

- (1) Die Masterarbeit ist fristgemäß in zweifacher Ausfertigung (maschinenschriftlich, gebunden und paginiert) sowie zusätzlich einmal in elektronischer Form durch ein physisches Medium beim Zentralen Prüfungssekretariat abzugeben; der Abgabezeitpunkt ist aktenkundig zu machen. Bei Zustellung der Arbeit mit der Post ist der Zeitpunkt der Einlieferung bei der Post (Poststempel) maßgebend. Wird die Masterarbeit nicht fristgemäß abgegeben, gilt sie als mit „mangelhaft“ (5,0) bewertet.

- (2) Die Masterarbeit ist von zwei Prüfenden gemäß § 9 zu bewerten. Der Vortrag der bzw. des Studierenden von ca. 45 Minuten Dauer geht in die Bewertung ein. Er findet spätestens vier Wochen nach dem Abgabzeitpunkt statt. Zu den Prüfenden soll insbesondere zählen, wer das Thema gestellt hat. Die bzw. der zweite Prüfende wird von der bzw. dem Vorsitzenden des Prüfungsausschusses bestimmt. Die Kandidatin bzw. der Kandidat hat ein Vorschlagsrecht. Dies begründet jedoch keinen Rechtsanspruch. Die einzelne Bewertung ist entsprechend § 16 Abs. 1 bis 3 vorzunehmen und schriftlich zu begründen. Die Note für die Arbeit wird aus dem arithmetischen Mittel der Einzelbewertungen gebildet, sofern die Differenz nicht mehr als 1,0 beträgt und die Noten der Einzelbewertungen jeweils mindestens „ausreichend“ sind. Beträgt die Differenz mehr als 1,0 oder lautet eine Bewertung „mangelhaft“, die andere aber mindestens „ausreichend“, wird vom Prüfungsausschuss eine dritte Prüferin bzw. ein dritter Prüfer zur Bewertung der Masterarbeit bestimmt. In diesem Fall wird die Note der Arbeit aus dem arithmetischen Mittel der drei Noten gebildet. Die Arbeit kann jedoch nur dann als „ausreichend“ oder besser bewertet werden, wenn mindestens zwei Noten „ausreichend“ oder besser sind.
- (3) Die Note der Masterarbeit ist gleichzeitig die Note des Moduls Master-Abschlussarbeit.
- (4) Die Bewertung der Masterarbeit ist den Studierenden in der Regel spätestens nach sechs Wochen nach Abgabe der Masterarbeit im Campus Management System der Universität Paderborn bekannt zu geben.

## § 19

### Bewertung der Masterprüfung und Bildung der Gesamtnote

- (1) Die Masterprüfung ist bestanden, wenn alle Modulprüfungen und die Masterarbeit mit mindestens „ausreichend“ (4,0) bewertet sind.
- (2) Die Gesamtnote wird gebildet, indem alle Modulnoten nach Leistungspunkten gewichtet werden. Dabei werden abweichend von den in § 10 Abs. 3 festgelegten LP das Modul Projektgruppe mit 8 Gewichtspunkten, das Modul Master-Abschlussarbeit mit 50 Gewichtspunkten und das Modul „Studium Generale“ mit 4 Gewichtspunkten gewichtet.

Bei der Berechnung des Ergebnisses wird nur die erste Dezimalstelle hinter dem Komma berücksichtigt, alle weiteren Stellen werden ohne Rundung gestrichen.

Die Note lautet:

bei einem Durchschnitt bis einschließlich 1,5 = sehr gut

bei einem Durchschnitt über 1,5 bis einschließlich 2,5 = gut

bei einem Durchschnitt über 2,5 bis einschließlich 3,5 = befriedigend

bei einem Durchschnitt über 3,5 bis einschließlich 4,0 = ausreichend

bei einem Durchschnitt über 4,0 bis 5,0 = mangelhaft

- (3) Das Gesamturteil „mit Auszeichnung bestanden“ wird erteilt, wenn die Gesamtnote 1,3 oder besser lautet.

## § 20

### Wiederholung von Prüfungsleistungen, Kompensation

- (1) Bestandene Prüfungen können nicht wiederholt werden.

- (2) Jede nicht bestandene Modulprüfung kann zweimal wiederholt werden.
- (3) Eine bestandene Prüfung, die als Zusatzleistung nach § 21 verbucht ist, kann auf Wunsch der Kandidatin oder des Kandidaten gegen eine bestandene oder eine noch nicht bzw. endgültig nicht bestandene Prüfung ausgetauscht werden (Kompensation), wenn jene vom Grundsatz her an deren Stelle verbucht werden kann.
- (4) Ein Modul ist nicht bestanden, wenn eine Modulprüfung oder Modulteilprüfung nicht mehr wiederholt oder gemäß Abs. 3 kompensiert werden kann.
- (5) Wiederholungsprüfungen, bei deren endgültigem Nichtbestehen keine Ausgleichsmöglichkeit vorgesehen ist, sind von mindestens zwei Prüfenden gemäß § 9 zu bewerten.
- (6) Das Modul Master-Abschlussarbeit kann bei einer Bewertung mit „mangelhaft“ (5,0) einmal wiederholt werden. Dabei ist ein neues Thema zu stellen. Eine zweite Wiederholung ist ausgeschlossen. Eine Rückgabe des Themas der Masterarbeit in der in § 17 Abs. 6 Satz 2 genannten Frist ist jedoch nur zulässig, wenn von der Rückgabemöglichkeit beim ersten Versuch kein Gebrauch gemacht wurde.
- (7) Für die Wiederholung der Masterarbeit kann die Kandidatin bzw. der Kandidat eine andere Prüfende oder einen anderen Prüfenden vorschlagen. Dies begründet jedoch keinen Rechtsanspruch.

## § 21 Zusatzmodule

- (1) Über die in § 10 geforderten Leistungen hinaus können Studierende Prüfungen zu Modulen im Umfang von bis zu 18 LP ablegen. Regelungen zu teilnehmerbegrenzten Modulen gemäß § 59 HG bleiben unberührt. Die erfolgreich abgeschlossenen Module werden im „Transcript of Records“ aufgeführt, es sei denn, dass die bzw. der Studierende deren Nichtaufführung bis zur Abgabe der Abschlussarbeit beantragt. Die werden bei der Notenbildung der Masterprüfung nicht berücksichtigt.
- (2) Unter Beachtung der in Absatz 1 angegebenen Obergrenze ist auch ein Umbuchen zum Zwecke einer Kompensation nach § 20 Abs. 3 möglich. Unter die Obergrenze fallen auch nicht bestandene Prüfungen.
- (3) Die Zusatzmodule sind als solche bei der Anmeldung zu kennzeichnen.

## § 22 Versäumnis, Rücktritt, Täuschung, Ordnungsverstoß, Schutzvorschriften und Studierende mit Familienaufgaben

- (1) Eine Abmeldung von Prüfungen kann bis spätestens eine Woche vor dem jeweiligen Prüfungstermin ohne Angabe von Gründen vorgenommen werden. Eine Abmeldung von Individualprüfungen kann bis spätestens eine Woche vor dem jeweiligen Prüfungstermin über das Campus Management System ohne Angabe von Gründen vorgenommen werden. Eine Abmeldung entsprechend Satz 1 kann bei Blockprüfungen nur bis spätestens eine Woche vor Beginn dieses Prüfungsblocks vorgenommen werden. Die Abmeldetermine für die Projektgruppe und Seminare werden von der bzw. dem Lehrenden vor Beginn der Prüfungsanmeldephase festgelegt.
- (2) Eine Prüfungsleistung gilt als mit „mangelhaft“ (5,0) bewertet, wenn die Kandidatin bzw. der Kandidat zu einem Prüfungstermin ohne triftige Gründe nicht erscheint oder nach Beginn der Prüfung ohne triftige Gründe von der Prüfung zurücktritt oder nach Ablauf der Abmeldefristen nach Abs. 3 ohne Angabe von triftigen Gründen von der Prüfung zurücktritt. Dasselbe gilt, wenn eine schriftliche Prüfungsleistung nicht innerhalb der vorgegebenen Bearbeitungszeit erbracht wird.

- (3) Die für das Versäumnis oder den Rücktritt geltend gemachten Gründe müssen dem Prüfungsausschuss unverzüglich, spätestens aber fünf Werktage nach dem jeweiligen Prüfungstermin schriftlich angezeigt und glaubhaft gemacht werden. Bei Krankheit der Kandidatin bzw. des Kandidaten reicht eine spätestens vom Tag der Prüfung datierte ärztliche Bescheinigung über das Bestehen der Prüfungsunfähigkeit aus. Bestehen zureichende tatsächliche Anhaltspunkte, die eine Prüfungsfähigkeit als wahrscheinlich annehmen oder einen anderen Nachweis als sachgerecht erscheinen lassen, kann eine ärztliche Bescheinigung einer Vertrauensärztin oder eines Vertrauensarztes der Universität Paderborn auf Kosten der Universität verlangt werden. Die durch ärztliche Bescheinigung belegte Erkrankung des Kindes im Sinne des § 25 Abs. 5 Bundesausbildungsförderungsgesetz gilt als Prüfungsunfähigkeit der Kandidatin bzw. des Kandidaten, wenn die Betreuung nicht anders gewährleistet werden konnte, insbesondere bei überwiegend alleiniger Betreuung. Erkennt der Prüfungsausschuss die Gründe an, wird der Kandidatin bzw. dem Kandidaten dies schriftlich mitgeteilt und ein neuer Prüfungstermin festgesetzt. Die bereits vorliegenden Prüfungsergebnisse sind in diesem Fall anzurechnen. Erkennt der Prüfungsausschuss die Gründe nicht an, wird dies der Kandidatin bzw. dem Kandidaten schriftlich mitgeteilt.
- (4) Täuscht eine Kandidatin bzw. ein Kandidat oder versucht sie bzw. er zu täuschen, gilt die betreffende Prüfungsleistung als mit „mangelhaft“ (5,0) bewertet. Führt eine Kandidatin bzw. ein Kandidat ein nicht zugelassenes Hilfsmittel mit sich, kann die betreffende Prüfungsleistung als mit „mangelhaft“ bewertet werden. Die Vorfälle werden von den jeweils Aufsichtsführenden aktenkundig gemacht. Die Feststellung gem. Satz 1 bzw. die Entscheidung gem. Satz 2 wird von der bzw. dem jeweiligen Prüfenden getroffen.
- (5) Eine Kandidatin bzw. ein Kandidat, die bzw. der den ordnungsgemäßen Ablauf der Prüfung stört, kann von den jeweiligen Prüfenden oder Aufsichtsführenden in der Regel nach Abmahnung von der Fortsetzung der jeweiligen Prüfungsleistung ausgeschlossen werden; in diesem Fall gilt die betreffende Prüfungsleistung als mit „mangelhaft“ (5,0) bewertet. Die Gründe für den Ausschluss sind aktenkundig zu machen.
- (6) In schwerwiegenden Fällen von Täuschung oder Störung kann der Prüfungsausschuss die Kandidatin bzw. den Kandidaten von weiteren Prüfungsleistungen ausschließen. Täuschungshandlungen können gem. § 63 Abs. 5 HG außerdem mit einer Geldbuße von bis zu 50.000 € geahndet werden und zur Exmatrikulation führen.
- (7) Die Kandidatin bzw. der Kandidat kann innerhalb von 14 Tagen verlangen, dass Entscheidungen nach Abs. 4 Satz 1 und 2 und Abs. 5 vom Prüfungsausschuss überprüft werden. Belastende Entscheidungen des Prüfungsausschusses sind der Kandidatin bzw. dem Kandidaten unverzüglich schriftlich mitzuteilen, zu begründen und mit einer Rechtsbehelfsbelehrung zu versehen. Vor der Entscheidung ist der Kandidatin bzw. dem Kandidaten Gelegenheit zum rechtlichen Gehör zu geben.
- (8) Außerdem regelt der Prüfungsausschuss den Nachteilsausgleich für Studierende mit Behinderung oder chronischer Erkrankung. Ist die bzw. der Studierende aufgrund ihrer bzw. seiner Behinderung oder chronischen Erkrankung nicht in der Lage, Leistungen ganz oder teilweise entsprechend der vorgesehenen Modalitäten zu erbringen, soll ein Nachteilsausgleich gewährt werden. Als Nachteilsausgleich kommen insbesondere die Gewährung von organisatorischen Maßnahmen und Hilfsmitteln, die Verlängerung der Bearbeitungszeit oder die Gestattung einer anderen, gleichwertigen Leistungserbringungsform in Betracht. Die Behinderung oder chronische Erkrankung ist glaubhaft zu machen. Hierzu kann ein ärztliches Attest oder psychologisches Gutachten verlangt werden. Der Antrag soll die gewünschten Modifikationen benennen und begründen. Auf Antrag der bzw. des Studierenden oder des Prüfungsausschusses im Einvernehmen mit der bzw. dem Studierenden kann die bzw. der Beauftragte für Studierende mit Behinderung oder chronischer Erkrankung Empfehlungen für die Gestaltung des Nachteilsausgleichs abgeben.

- (9) Der besonderen Situation von Studierenden mit Familienaufgaben beim Studium und bei der Erbringung von Leistungen wird Rechnung getragen. Dies geschieht unter anderem in folgenden Formen:
- a. Auf Antrag einer Kandidatin sind die Schutzbestimmungen gem. §§ 3, 4, 6 und 8 des Mutterschutzgesetzes entsprechend zu berücksichtigen. Dem Antrag sind die erforderlichen Nachweise beizufügen. Der Prüfungsausschuss kann unter Berücksichtigung des Einzelfalls andere Leistungserbringungsformen festlegen. Die Mutterschutzfristen unterbrechen jede Frist nach dieser Prüfungsordnung; die Dauer des Mutterschutzes wird nicht in die Frist eingerechnet.
  - b. Gleichfalls sind die Fristen der Elternzeit nach Maßgabe des jeweils gültigen Bundeselterngeld- und Elternzeitgesetzes auf Antrag zu berücksichtigen. Die Kandidatin bzw. der Kandidat muss bis spätestens vier Wochen vor dem Zeitpunkt, ab dem sie bzw. er die Elternzeit antreten will, dem Prüfungsausschuss unter Beifügung der erforderlichen Nachweise schriftlich mitteilen, für welchen Zeitraum oder für welche Zeiträume sie bzw. er eine Elternzeit in Anspruch nehmen will. Der Prüfungsausschuss prüft, ob die gesetzlichen Voraussetzungen vorliegen, die bei einer Arbeitnehmerin bzw. einem Arbeitnehmer einen Anspruch auf Elternzeit nach dem Bundeselterngeld- und Elternzeitgesetz auslösen würden und legt unter Berücksichtigung des Einzelfalls die Termine und Fristen fest. Die Abgabefrist der Masterarbeit kann höchstens auf das Doppelte der vorgesehenen Bearbeitungszeit verlängert werden. Andernfalls gilt die gestellte Arbeit als nicht vergeben und die Kandidatin bzw. der Kandidat erhält nach Ablauf der Elternzeit ein neues Thema.
  - c. Der Prüfungsausschuss berücksichtigt auf Antrag Ausfallzeiten durch die Pflege und Erziehung von Kindern im Sinne des § 25 Absatz 5 Bundesausbildungsförderungsgesetz und Ausfallzeiten durch die Pflege der Ehegattin bzw. des Ehegatten, der eingetragenen Lebenspartnerin bzw. des eingetragenen Lebenspartners, der Partnerin bzw. des Partners einer eheähnlichen Gemeinschaft oder eines in gerader Linie Verwandten oder ersten Grades Verschwägerten und legt unter Berücksichtigung des Einzelfalls die Fristen und Termine fest. Im Übrigen gelten die Sätze 4 und 5 von Buchstabe b) entsprechend.

## § 23

### Erfolgreicher Abschluss des Studiums, endgültiges Nichtbestehen

- (1) Das Studium ist erfolgreich abgeschlossen, wenn die Masterprüfung bestanden ist und alle Module erfolgreich abgeschlossen sind. Die Masterprüfung ist bestanden, wenn alle Modulprüfungen sowie die Prüfungen im optionalen Nebenfach oder im Studium Generale mindestens mit der Note „ausreichend“ (4,0) benotet wurden.
- (2) Die Masterprüfung ist endgültig nicht bestanden, wenn ein Modul endgültig nicht bestanden ist oder das Modul Master-Abschlussarbeit zum zweiten Mal mit der Note „mangelhaft“ bewertet wird.
- (3) Der Bescheid über eine endgültig nicht bestandene Masterprüfung wird der Kandidatin bzw. dem Kandidaten durch die Vorsitzende bzw. den Vorsitzenden des Prüfungsausschusses in schriftlicher Form erteilt. Der Bescheid ist mit einer Rechtsbehelfsbelehrung zu versehen.
- (4) Hat die Kandidatin oder der Kandidat die Masterprüfung endgültig nicht bestanden, wird ihr bzw. ihm auf Antrag ein Leistungszeugnis ausgestellt, das die erbrachten Leistungen und gegebenenfalls die erworbenen Leistungspunkte enthält und das erkennen lässt, dass die Masterprüfung endgültig nicht bestanden ist.
- (5) Studierenden, die die Hochschule aus anderen Gründen ohne Hochschulabschluss verlassen, ist nach der Exmatrikulation auf Antrag ein Leistungszeugnis auszustellen, das die erbrachten Leistungen und gegebenenfalls die erworbenen Leistungspunkte enthält.

## § 24 Zeugnis, Transcript of Records, Diploma Supplement

- (1) Hat die Kandidatin bzw. der Kandidat das Studium erfolgreich abgeschlossen, erhält sie bzw. er über das Ergebnis ein Zeugnis. Dieses Zeugnis enthält den Namen des Studienganges, die Specialization Area, die Regelstudienzeit und die Gesamtnote. Das Zeugnis weist das Datum auf, an dem die letzte Prüfungsleistung erbracht worden ist. Daneben trägt es das Datum der Ausfertigung. Das Zeugnis ist von der bzw. dem Vorsitzenden des Prüfungsausschusses zu unterzeichnen.
- (2) Ferner erhält die Kandidatin bzw. der Kandidat ein Transcript of Records, in dem die gesamten erbrachten Leistungen und die Fachstudiendauer aufgeführt sind. Das Transcript of Records enthält Angaben über die Leistungspunkte und die erzielten Noten zu den absolvierten Modulen und zu der Masterarbeit. Es enthält des Weiteren das Thema der Masterarbeit und die erzielte Gesamtnote der Masterprüfung.
- (3) Mit dem Zeugnis wird der Absolventin bzw. dem Absolventen ein Diploma Supplement ausgehändigt.
- (4) Das Diploma Supplement ist eine Zeugnisergänzung in englischer und deutscher Sprache mit einheitlichen Angaben zu den deutschen Hochschulabschlüssen, welche das deutsche Bildungssystem erläutern und die Einordnung des vorliegenden Abschlusses vornimmt. Das Diploma Supplement informiert über den absolvierten Studiengang und die mit dem Abschluss erworbenen akademischen und beruflichen Qualifikationen. Es enthält die wesentlichen dem Abschluss zugrunde liegenden Studieninhalte, den Studienverlauf, die mit dem Abschluss erworbenen Kompetenzen sowie die verleihende Hochschule.
- (5) Falls nach Maßgabe von § 10 Abs. 5 Prüfungsleistungen in ausreichendem Umfang in englischer Sprache abgelegt worden sind, wird der Abschluss „Englischsprachiger Masterstudiengang Informatik“ auf dem Zeugnis und Transcript of Records bescheinigt.

## § 25 Masterurkunde

- (1) Gleichzeitig mit dem Zeugnis über den bestandenen Masterabschluss wird der Kandidatin bzw. dem Kandidaten die Masterurkunde mit dem Datum des Zeugnisses ausgehändigt. Darin wird die Verleihung des Mastergrades gemäß § 2 beurkundet.
- (2) Die Masterurkunde wird von der bzw. dem Vorsitzenden des Prüfungsausschusses und der Dekanin bzw. dem Dekan der Fakultät für Elektrotechnik, Informatik und Mathematik unterzeichnet und mit dem Siegel der Universität Paderborn versehen.
- (3) Der Masterurkunde wird eine englischsprachige Übersetzung beigelegt.

## § 26 Einsicht in die Prüfungsakten

- (1) Der Kandidatin bzw. dem Kandidaten kann die Möglichkeit gegeben werden, nach Bekanntgabe der Noten Einsicht in ihre bzw. seine schriftlichen Prüfungsleistungen und die darauf bezogenen Bewertungen der Prüfenden zu nehmen. Die bzw. der Prüfende bestimmt Ort und Zeit der Einsichtnahme und gibt diese in geeigneter Form bekannt.

- (2) Sofern Absatz 1 nicht angewendet wird, wird der Kandidatin bzw. dem Kandidaten auf Antrag bis spätestens einen Monat nach Bekanntgabe der Ergebnisse der jeweiligen Prüfungen Einsicht in ihre bzw. seine schriftlichen Prüfungsleistungen, die darauf bezogenen Bewertungen der Prüfenden und in die Prüfungsprotokolle gewährt. Innerhalb eines Jahres nach Aushändigung des Zeugnisses wird der Kandidatin bzw. dem Kandidaten auf Antrag in angemessener Frist Einsicht in die Masterarbeit, die darauf bezogenen Gutachten der Prüfenden und in die Prüfungsprotokolle gewährt. Die bzw. der Vorsitzende des Prüfungsausschusses bestimmt Ort und Zeit der Einsichtnahme; sie bzw. er kann diese Aufgaben an die Prüfenden delegieren.

## **IV. Schlussbestimmungen**

### **§ 27**

#### **Ungültigkeit der Masterprüfung**

- (1) Hat die Kandidatin bzw. der Kandidat bei einer Prüfung getäuscht und wird diese Tatsache erst nach Aushändigung des Zeugnisses bekannt, kann der Prüfungsausschuss nachträglich die Noten für diejenigen Prüfungsleistungen, bei deren Erbringung die Kandidatin bzw. der Kandidat getäuscht hat, 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 Kandidatin bzw. der Kandidat hierüber täuschen wollte, und wird diese Tatsache erst nach der Aushändigung des Zeugnisses bekannt, wird dieser Mangel durch das Bestehen der Prüfung geheilt. Hat die Kandidatin bzw. der Kandidat die Zulassung vorsätzlich zu Unrecht erwirkt, entscheidet der Prüfungsausschuss unter Beachtung des Verwaltungsverfahrensgesetzes für das Land Nordrhein-Westfalen über die Rechtsfolgen.
- (3) Vor einer Entscheidung ist der oder dem Betroffenen Gelegenheit zur Äußerung zu geben.
- (4) Das unrichtige Zeugnis ist einzuziehen und gegebenenfalls ein neues zu erteilen. Eine Entscheidung nach Absatz 1 und Absatz 2 Satz 2 ist nach einer Frist von fünf Jahren nach Ausstellung des Prüfungszeugnisses ausgeschlossen.
- (5) Ist die Prüfung insgesamt für nicht bestanden erklärt worden, sind der Mastergrad abzuerkennen und die entsprechende Urkunde einzuziehen. Die Aberkennung des Mastergrades ist nur innerhalb von fünf Jahren seit dem Zeitpunkt der Gradverleihung zulässig.

### **§ 28**

#### **Aberkennung des Mastergrades**

Der Mastergrad kann aberkannt werden, wenn sich nachträglich herausstellt, dass er durch Täuschung erworben worden ist, oder wenn wesentliche Voraussetzungen für die Verleihung irrtümlich als gegeben angesehen worden sind. Über die Aberkennung entscheidet der Fakultätsrat mit zwei Dritteln seiner Mitglieder. Die Aberkennung ist nur innerhalb von fünf Jahren seit dem Zeitpunkt der Gradverleihung zulässig.

## § 29 Übergangsbestimmungen

- (1) Diese Prüfungsordnung gilt für alle Studierenden, die erstmalig ab dem Wintersemester 2017/18 für den Masterstudiengang Informatik der Fakultät für Elektrotechnik, Informatik und Mathematik an der Universität Paderborn eingeschrieben werden.
- (2) Studierende, die vor dem Wintersemester 2017/2018 an der Universität Paderborn für den Masterstudiengang Informatik eingeschrieben worden sind, können ihre Masterprüfung einschließlich Wiederholungsprüfungen letztmalig im Sommersemester 2020 nach der Prüfungsordnung in der Fassung vom 28. Februar 2013 (AM.Uni.Pb 11.13), geändert durch Satzung vom 29. Januar 2016 (AM.Uni.Pb 06.16), ablegen. Engere Fristen aus älteren Übergangsbestimmungen bleiben unberührt. Ab dem Wintersemester 2020/2021 wird die Masterprüfung einschließlich Wiederholungsprüfungen nach dieser Prüfungsordnung abgelegt.
- (3) Auf Antrag können Studierende in diese Prüfungsordnung wechseln. Der Wechsel ist unwiderruflich.

## § 30 Inkrafttreten und Veröffentlichung

- (1) Diese Prüfungsordnung tritt mit Wirkung vom 1. Oktober 2017 in Kraft. Gleichzeitig tritt die Masterprüfungsordnung vom 28. Februar 2013 (AM.Uni.Pb 11.13), geändert durch Satzung vom 29. Januar 2016 (AM.Uni.Pb 06.16), außer Kraft. § 29 bleibt unberührt.
- (2) Diese Prüfungsordnung wird in den Amtlichen Mitteilungen der Universität Paderborn (AM.Uni.Pb.) veröffentlicht.

Ausgefertigt aufgrund des Beschlusses des Fakultätsrates der Fakultät für Elektrotechnik, Informatik und Mathematik vom 24. April 2017 und der Rechtmäßigkeitsprüfung durch das Präsidium der Universität Paderborn vom 17. Mai 2017.

Paderborn, den 16. Juni 2017

Für den Präsidenten  
Die Vizepräsidentin für Wirtschafts- und Personalverwaltung  
der Universität Paderborn

Simone Probst



## Anhang 1: Module und Prüfungsformen

Als Folge der Weiterentwicklung der Forschungs- und Lehrinhalte der Institute für Informatik und für Elektrotechnik und Informationstechnik können im Wahlpflichtbereich Module der nachfolgenden Liste in geringer Zahl entfallen oder durch Module, die fachlich zu dem gleichen Bereich gehören, in geringer Zahl ersetzt oder ergänzt werden. Die Änderungen werden im Modulhandbuch bekannt gegeben. Die Regelungen zu den Leistungen, zum Umfang sowie zu Teilnahmevoraussetzungen bleiben hiervon unberührt.

Focus Area Modul	LP Modul SWS	Prüfungsformen	Bemerkung
<b>Software Engineering</b>	<b>6</b>	Mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul  <b>Im Falle der Einschreibung mit Auflagen</b> mit den Auflagenkursen <b>Grundlagen Mensch-Maschine-Wechselwirkung Software Engineering 1 und 2</b> ist das Bestehen der zugehörigen Prüfungen bis zur Anmeldung des Moduls nachzuweisen. Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Advanced Compiler Construction</li> <li>• Advanced Software Engineering: Methods, Architectures, Industrial Applications</li> <li>• Build It, Break It, Fix It</li> <li>• Compiler Construction</li> <li>• Deductive Verification</li> <li>• Designing code analyses for large-scale software systems</li> <li>• Empiric Performance Evaluation</li> <li>• Fundamentals of Model-Driven Engineering</li> <li>• High-Performance Computing</li> <li>• Kontextuelle Informatik</li> <li>• Language-Based Security</li> <li>• Logic Programming for Artificial Intelligence</li> <li>• Model Checking</li> <li>• Model-Based Interface Development</li> <li>• Software Analysis</li> <li>• Software Quality Assurance</li> <li>• Type Systems for Correctness and Security</li> <li>• Usability Engineering PracticeDeductive</li> </ul>	3+2		
<b>Algorithm Design</b>	<b>6</b>	Mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul  Im Falle der Einschreibung mit Auflagen mit den Auflagenkursen <b>Mathematik 1 und 2 Modelle und Algorithmen 1 und 2</b> ist das Bestehen der zugehörigen Prüfungen bis
Eines der Module <ul style="list-style-type: none"> <li>• Advanced Algorithms</li> <li>• Advanced Complexity Theory</li> <li>• Advanced Distributed Algorithms and Data Structures</li> <li>• Algorithmic Game Theory</li> <li>• Algorithms for Highly Complex Virtual Scenes</li> </ul>	3+2		

Focus Area Modul	LP Modul SWS	Prüfungsformen	Bemerkung
<ul style="list-style-type: none"> <li>• Clustering Algorithms</li> <li>• Computational Geometry</li> <li>• Foundations of Cryptography</li> <li>• Linear and Integer Optimization</li> <li>• Public-Key Cryptography</li> <li>• Routing and Data Management in Networks</li> </ul>			zur Anmeldung des Moduls nachzuweisen. Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
<b>Networks and Communication</b>	<b>6</b>	Mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul  Im Falle der Einschreibung mit Auflagen mit den Auflagenkursen <b>Mathematik 1 und 2</b> <b>Systems 1 und 2</b> ist das Bestehen der zugehörigen Prüfungen bis zur Anmeldung des Moduls nachzuweisen. Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Advanced Distributed Algorithms and Data Structures</li> <li>• Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies</li> <li>• Empiric Performance Evaluation</li> <li>• Future Internet</li> <li>• Mobile Communication</li> <li>• Network Simulation</li> <li>• Networked Embedded Systems</li> <li>• Routing and Data Management in Networks</li> <li>• Vehicular Networking</li> </ul>	3+2		
<b>Computer Systems</b>	<b>6</b>	Mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul  Im Falle der Einschreibung mit Auflagen mit den Auflagenkursen <b>Systems 1 und 2</b> ist das Bestehen der zugehörigen Prüfungen bis zur Anmeldung des Moduls nachzuweisen. Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Adaptive Hardware and Systems</li> <li>• Advanced Computer Architecture</li> <li>• Algorithms for Synthesis and Optimization of Integrated Circuits</li> <li>• Architectures of Parallel Computer Systems</li> <li>• Empiric Performance Evaluation</li> <li>• Hardware/Software Codesign</li> <li>• High-Performance Computing</li> <li>• Intelligence in Embedded Systems</li> <li>• Reconfigurable Computing</li> <li>• VLSI Testing</li> </ul>	3+2		
<b>Intelligence and Data</b>	<b>6</b>	Mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul  Im Falle der Einschreibung mit Auflagen mit den Auflagenkursen <b>Algorithmen und Modelle 1 und 2</b> <b>Mathematik 1 und 2</b> <b>Software Engineering 1 und 2</b> ist das Bestehen der zugehörigen Prüfungen bis
Eines der Module <ul style="list-style-type: none"> <li>• Clustering Algorithms</li> <li>• Intelligence in Embedded Systems</li> <li>• Interactive Data Visualization</li> <li>• Logic and Automated Reasoning</li> <li>• Logic Programming for Artificial Intelligence</li> <li>• Machine Learning I</li> <li>• Machine Learning II</li> </ul>	3+2		

Focus Area Modul	LP Modul SWS	Prüfungsformen	Bemerkung
<ul style="list-style-type: none"> <li>Planning and Heuristic Search</li> </ul>			zur Anmeldung des Moduls nachzuweisen. Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
<i>Projektgruppe</i>	20	Modulprüfung der Projektarbeit als phasenbezogene Prüfung	Pflichtmodul  Im Falle der Einschreibung mit Auflagen mit den Auflagenkursen <b>Soft Skills, Management</b> ist das Bestehen der zugehörigen Prüfungen bis zur Anmeldung des Moduls nachzuweisen. Qualifizierte Teilnahme beim Praktikum
<i>Seminar I</i>	5	Seminarvortrag und schriftliche Ausarbeitung, Note ergibt die Modulnote	Pflichtmodul  Im Falle der Einschreibung mit Auflagen mit dem Auflagenkurs <b>Soft Skills, Management</b> ist das Bestehen der zugehörigen Prüfungen bis zur Anmeldung des Moduls nachzuweisen.
<i>Seminar II</i>	Identisch zu Seminar I		
<i>Nebenfach oder Studium Generale</i>	12		Es wird entweder ein Nebenfach oder Studium Generale studiert (Ausnahme Wirtschaftsinformatik und Wirtschaftswissenschaften)
Nebenfach	12	Modulabschlussprüfung gemäß der Nebenfachvereinbarung	Wählbares Nebenfach aus dem Katalog der Standardnebenfächer bzw. Nicht-Standardnebenfach auf individuellen Antrag
Studium Generale	12	Prüfung im Studium Generale	
<i>Master-Abschlussarbeit</i>	30	Prüfung gemäß §17 und §18	Bestehend aus Arbeitsplanung, Anfertigung der Masterarbeit einschließlich Abschlussvortrag

Bei einer **Zulassung mit Auflagen** gemäß § 4, Abschnitt c, müssen gegebenenfalls Auflagenkurse absolviert werden, bevor ein Modul begonnen werden kann. Die folgenden Auflagenkurse sind als Voraussetzungen für Module geplant:

<b>Auflagenkurs</b>	<b>Umfang</b>	<b>Voraussetzung für Module der Focus Area</b>
<i>Grundlagen Mensch-Maschine-Wechselwirkung</i>	2 SWS	Software Engineering
<i>Mathematik 1 und 2</i>	4 SWS	Algorithm Design Networks and Communication Intelligence and Data
<i>Modelle und Algorithmen 1 und 2</i>	4 SWS	Algorithm Design Intelligence and Data
<i>Soft Skills, Management</i>	2 SWS	Seminar I und II Projektgruppe
<i>Software Engineering 1 und 2</i>	4 SWS	Software Engineering Intelligence and Data
<i>Systems 1 und 2</i>	4 SWS	Networks and Communication Computer Systems

## Anhang 2: Nebenfachvereinbarungen für die Standardnebenfächer im Masterstudiengang Informatik

Vorbemerkung: Die näheren Prüfungsmodalitäten bestimmen sich nach den Regelungen der Prüfungsordnung des jeweils einschlägigen Studiengangs in der jeweils geltenden Fassung.

Legende:

MP = Modulprüfung

SP = qualifizierte Teilnahme nachgewiesen durch Erstellung eines Seminarpapiers

qT = qualifizierte Teilnahme

### 1. Elektrotechnik

1.-3. Semester	2 Module aus <ul style="list-style-type: none"> <li>• Schaltungstechnik</li> <li>• Regelungstechnik</li> <li>• Nachrichtentechnik</li> <li>• Energietechnik</li> </ul>	MP	6
		MP	6
		<b>Summe</b>	<b>12</b>

### 2. Mathematik

1.-3. Semester	Weiterführende Module aus dem Kanon der Mathematikveranstaltungen	MP	9
		<b>Summe</b>	<b>9</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale zusätzlich 3 LP absolviert werden.

### 3. Medienwissenschaft

1.-3. Semester	<b>Basismodul Medientheorie/-geschichte</b>		
	Einführung	SP	4
	Seminar/Lehrveranstaltung	SP	4
	Seminar/Lehrveranstaltung	MP	4
		<b>SUMME</b>	<b>12</b>

### 4. Philosophie

1.-3. Semester	Aufbaumodul 1: Anthropologie und Kulturphilosophie	<b>jeweils:</b> Überblicksveranstaltung	<b>jeweils:</b> qT	<b>jeweils:</b> 12
----------------	---	--	-----------------------	--------------------

	<p style="text-align: center;"><b>oder</b></p> <p>Aufbaumodul 2: Praktische Philosophie</p> <p style="text-align: center;"><b>oder</b></p> <p>Aufbaumodul 3: Theoretische Philosophie</p>	<p>Seminar Seminar</p> <p>Modulprüfung</p>	<p>qT qT</p> <p>MP</p>	
			<b>SUMME</b>	<b>12</b>

### 5. Psychologie

1., 2. und 3. Semester	Forschungskolloquium Kognitionspsychologie	qT	2
	2 Seminare aus dem Lehrangebot der Teilgebiete Kognitionspsychologie und Arbeits- und Organisationspsychologie	qT	4
		qT	4
	Portfolioprüfung	MP	2
		<b>SUMME</b>	<b>12</b>

### 6. Wirtschaftsinformatik

1., 2. und 3. Semester	Alle Module aus dem Modulkatalog der Fakultät für Wirtschaftswissenschaften mit der Nummer M.184.43XX	MP	10
		<b>SUMME</b>	<b>10</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale zusätzlich 2 LP absolviert werden.

### 7. Wirtschaftswissenschaften

1., 2. und 3. Semester	Alle Module aus dem Modulkatalog der Fakultät für Wirtschaftswissenschaften außer mit der Nummer M.184.43XX	MP	10
		<b>SUMME</b>	<b>10</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale zusätzlich 2 LP absolviert werden.

## Anhang 3: Studienverlaufsplan

Studienplan Informatik Master (exemplarisch)					
1. Semester	Wahlpflicht- modul I (6 LP)	Wahlpflicht- modul II (6 LP)	Wahlpflicht- modul III (6 LP)	Wahlpflicht- modul IV (6 LP)	Wahlpflicht- modul V (6 LP)
2. Semester	Projektgruppe (20 LP)	Wahlpflicht- modul VI (6 LP)		Seminar I (5 LP)	SG/NF
3. Semester		Wahlpflicht- modul VII (6 LP)	Wahlpflicht- modul VIII (6 LP)	Seminar II (5 LP)	SG/NF
4. Semester	Masterarbeit (30 LP)				

## Anhang 4: Modulhandbuch

# Inhaltsverzeichnis

<b>1</b>	<b>Beschreibung des Studiengangs Master Informatik</b>	<b>4</b>
1.1	Schema der Modulbeschreibungen . . . . .	5
1.2	Liste der Organisationsformen . . . . .	7
1.3	Liste der Prüfungsformen . . . . .	7
1.4	Liste der nichtkognitive Kompetenzen . . . . .	8
<b>2</b>	<b>Studienrichtungen</b>	<b>10</b>
2.1	Algorithm Design . . . . .	11
2.2	Computer Systems . . . . .	12
2.3	Intelligence and Data . . . . .	13
2.4	Networks and Communication . . . . .	14
2.5	Software Engineering . . . . .	15
<b>3</b>	<b>Module</b>	<b>17</b>
3.1	Wahlpflichtmodul: Adaptive Hardware and Systems . . . . .	18
3.2	Wahlpflichtmodul: Advanced Algorithms . . . . .	21
3.3	Wahlpflichtmodul: Advanced Compiler Construction . . . . .	23
3.4	Wahlpflichtmodul: Advanced Complexity Theory . . . . .	25
3.5	Wahlpflichtmodul: Advanced Computer Architecture . . . . .	28
3.6	Wahlpflichtmodul: Advanced Distributed Algorithms and Data Structures . . . . .	30
3.7	Wahlpflichtmodul: Advanced Software Engineering: Methods, Architectures, Industrial Applications . . . . .	32
3.8	Wahlpflichtmodul: Algorithmic Game Theory . . . . .	34
3.9	Wahlpflichtmodul: Algorithms for Highly Complex Virtual Scenes . . . . .	36
3.10	Wahlpflichtmodul: Algorithms for Synthesis and Optimization of Integrated Circuits . . . . .	39
3.11	Wahlpflichtmodul: Architektur paralleler Rechnersysteme . . . . .	41
3.12	Wahlpflichtmodul: Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies . . . . .	44
3.13	Wahlpflichtmodul: Build It, Break It, Fix It . . . . .	47
3.14	Wahlpflichtmodul: Clustering Algorithms . . . . .	50
3.15	Wahlpflichtmodul: Compiler Construction . . . . .	52
3.16	Wahlpflichtmodul: Computational Geometry . . . . .	54
3.17	Wahlpflichtmodul: Deductive Verification . . . . .	56
3.18	Wahlpflichtmodul: Designing code analyses for large-scale software systems . . . . .	58
3.19	Wahlpflichtmodul: Empiric performance evaluation . . . . .	61
3.20	Wahlpflichtmodul: Foundations of Cryptography . . . . .	63
3.21	Wahlpflichtmodul: Fundamentals of Model-Driven Engineering . . . . .	65
3.22	Wahlpflichtmodul: Future Internet . . . . .	68



3.23	Wahlpflichtmodul: Hardware/Software Codesign . . . . .	70
3.24	Wahlpflichtmodul: High-Performance Computing . . . . .	72
3.25	Wahlpflichtmodul: Intelligence in Embedded Systems . . . . .	74
3.26	Wahlpflichtmodul: Interactive Data Visualization . . . . .	76
3.27	Wahlpflichtmodul: Kontextuelle Informatik . . . . .	79
3.28	Wahlpflichtmodul: Language-Based Security . . . . .	82
3.29	Wahlpflichtmodul: Linear and Integer Optimization . . . . .	84
3.30	Wahlpflichtmodul: Logic and Automated Reasoning . . . . .	86
3.31	Wahlpflichtmodul: Logic Programming for Artificial Intelligence . . . . .	89
3.32	Wahlpflichtmodul: Machine Learning I . . . . .	92
3.33	Wahlpflichtmodul: Machine Learning II . . . . .	94
3.34	Pflichtmodul: Master-Abschlussarbeit . . . . .	96
3.35	Wahlpflichtmodul: Mobile Communication . . . . .	99
3.36	Wahlpflichtmodul: Model Checking . . . . .	101
3.37	Wahlpflichtmodul: Model-Based Interface Development . . . . .	103
3.38	Wahlpflichtmodul: Network Simulation . . . . .	106
3.39	Wahlpflichtmodul: Networked Embedded Systems . . . . .	108
3.40	Wahlpflichtmodul: Planning and Heuristic Search . . . . .	110
3.41	Pflichtmodul: Projektgruppe . . . . .	113
3.42	Wahlpflichtmodul: Public-Key Cryptography . . . . .	116
3.43	Wahlpflichtmodul: Reconfigurable Computing . . . . .	118
3.44	Wahlpflichtmodul: Routing and Data Management in Networks . . . . .	120
3.45	Pflichtmodul: Seminar I . . . . .	122
3.46	Pflichtmodul: Seminar II . . . . .	124
3.47	Wahlpflichtmodul: Software Analysis . . . . .	126
3.48	Wahlpflichtmodul: Software Quality Assurance . . . . .	128
3.49	Pflichtmodul: Studium Generale . . . . .	130
3.50	Wahlpflichtmodul: Type Systems for Correctness and Security . . . . .	132
3.51	Wahlpflichtmodul: Usability Engineering Practice . . . . .	135
3.52	Wahlpflichtmodul: Vehicular Networking . . . . .	138
3.53	Wahlpflichtmodul: VLSI Testing . . . . .	140
<b>A</b>	<b>Überblickstabellen</b>	<b>142</b>
A.1	Studienrichtungen und Module . . . . .	143
A.2	Module und Lehrveranstaltungen . . . . .	145

# Kapitel 1

## Beschreibung des Studiengangs Master Informatik

Ziel des viersemestrigen Master-Studienganges Informatik ist die Vertiefung der in einem Bachelor-Studiengang erworbenen Kenntnisse in einem oder mehreren Gebieten der Informatik. Dabei ist eines von fünf Gebieten als Vertiefungsgebiet (Focus Area) zu wählen - in diesem Gebiet müssen mindestens 3 Module (je 6 ECTS) studiert werden; außerdem muss die Masterarbeit in diesem Gebiet geschrieben werden. In mindestens einem weiteren Gebiet ist ein Modul zu absolvieren. Hinzu kommt eine Projektgruppe (20 ECTS), eine zweisemestrige Veranstaltungsform, in der im Team an einem forschungsnahen Thema gearbeitet wird. Im Übrigen können Studierenden ihr Studium frei organisieren.

## 1.1 Schema der Modulbeschreibungen

Die Modulbeschreibungen sind nach folgendem Schema einheitlich strukturiert:

Modulname	<Name des Moduls>
Workload	<Gesamtaufwand in Stunden (Workload ECTS)>
Leistungspunkte	<Gesamtaufwand in Leistungspunkten ECTS>
Studiensemester	<Liste der Lehrveranstaltungen in diesem Modul mit Zielsemester>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
<Liste der im Modul enthaltenen Lehrveranstaltungen mit Aufteilung des Workloads in Kontaktzeit und Selbststudium, Sprache in der die Veranstaltung gehalten wird, Winter- oder Sommersemester und ungefähre Gruppengröße.>	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
<Liste der im Modul enthaltenen Wahlmöglichkeiten.>	
<b>Teilnahmevoraussetzungen</b>	
<Voraussetzungen für die Teilnahme am Modul.>	
<b>Empfohlene Kenntnisse</b>	
<Die Angaben sind als Empfehlungen zu verstehen, nicht jedoch als zu überprüfende Voraussetzungen.>	
<b>Inhalte</b>	
<Aufzählung der wesentlichen Inhalte der enthaltenen Lehrveranstaltungen.>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<Aufzählung der erreichten Kenntnisse, Fähigkeiten und Fachkompetenzen.>	
<b>Nichtkognitive Kompetenzen</b>	
<Zusammenfassung aller nichtkognitiver Kompetenzen, die in den Lehrveranstaltungen des Moduls vermittelt werden.>	
<b>Methodische Umsetzung</b>	
<Angaben zu Sozialformen und didaktisch-methodischen Arbeitsweisen in den Veranstaltungen.>	
<b>Prüfungsleistung (Dauer)</b>	
<Form in Dauer der im Modul zu erbringenden Prüfungsleistung.>	
<b>Modulteilprüfungen</b>	
<Form der im Modul zu erbringenden Modulteilprüfung.>	

<b>Studienleistung / qualifizierte Teilnahme</b>
<Form der im Modul zu erbringenden Studienleistungen oder qualifizierter Teilnahmen.>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
<Formale Voraussetzungen für Teilnahme an der Modulprüfung.>
<b>Voraussetzungen für die Vergabe von Credits</b>
<Formale Voraussetzungen für die Vergabe von Credits.>
<b>Gewichtung für die Gesamtnote</b>
<Gesamtgewichtung des Moduls bei der Berechnung des Notendurchschnitts.>
<b>Modul wird in folgenden Studiengängen verwendet</b>
<Liste der Studiengänge, in denen dieses Modul verwendet wird.>
<b>Modulbeauftragte/r</b>
<Verantwortlicher für das Modul.>
<b>Lernmaterialien, Literaturangaben</b>
<Angaben zu Literatur, Vorlesungsskripten, etc.>
<b>Sonstige Hinweise</b>
<Sonstige Hinweise.>

## 1.2 Liste der Organisationsformen

Die folgenden Organisationsformen werden in diesem Studiengang verwendet:

### **Abschlussarbeit**

**Praktikum** In Kleingruppen arbeiten Studierende an praktischen Aufgaben.

### **Seminar**

**Vorlesung mit Übung** Eine Kombination aus Vorlesung und begleitenden Übungen, häufig mit praktischen Anteilen und Hausaufgaben.

**Vorlesung mit Übung und Praktikum** Eine Vorlesung mit Übungen wird mit einem Praktikumsteil kombiniert.

**Vorlesungen, Seminare, Projekte** Eine beliebige Kombination von Veranstaltungen für das Studium Generale.

## 1.3 Liste der Prüfungsformen

Die folgenden Organisationsformen werden in diesem Studiengang verwendet:

**Abschlussarbeit** Eine Abschlussarbeit soll zeigen, dass die Kandidatin oder der Kandidat die Fähigkeit besitzt, innerhalb einer bestimmten Frist ein Problem der Informatik auf der Grundlage wissenschaftlicher Methoden zu bearbeiten.

**Mündliche Prüfung** In den mündlichen Prüfungen soll die Kandidatin bzw. der Kandidat nachweisen, dass sie bzw. er die Zusammenhänge des Prüfungsgebietes erkennt und spezielle Fragestellungen in diese Zusammenhänge einzuordnen vermag. Durch die mündlichen Prüfungen soll ferner festgestellt werden, ob die Kandidatin bzw. der Kandidat über breites Grundlagenwissen verfügt. Mündliche Prüfungen dauern in der Regel mindestens 25 und höchstens 50 Minuten. Bei Gruppenprüfungen verlängert sich die Gesamtprüfungsdauer entsprechend der Kandidatenzahl.

**Phasenbezogene Prüfung (100 % der Modulnote)** In einer phasenbezogenen Prüfung ist die erfolgreiche Bearbeitung von Projekten durch die Abgabe von Software und Dokumentation nachzuweisen. Es wird eine Note für die Gesamtheit der bearbeiteten Projekte vergeben.

**Prüfung im Studium Generale** In einer der im Studium Generale gewählten Veranstaltungen wird eine mündliche oder schriftliche Prüfungsleistung erbracht. Dabei handelt es sich in der Regel um eine Klausur (maximal vier Stunden), eine Hausarbeit (maximal 25 Seiten) oder eine mündliche Prüfung (maximal 45 Minuten).

**Seminarvortrag (45-60 Minuten) und schriftliche Ausarbeitung** In einem Seminarvortrag sollen die Studierenden nachweisen, dass sie zur wissenschaftlichen Ausarbeitung eines Themas in der Lage sind und die Ergebnisse vortragen können. In der schriftlichen Ausarbeitung sollen sie zeigen, dass sie in der Lage sind Ergebnisse gemäß wissenschaftlichen Standards darzustellen.

**Studienleistung** Als Studienleistung können Übungsaufgaben verlangt werden, die in der Regel wöchentlich als Hausaufgaben und/oder Präsenzaufgaben gestellt werden. Studienleistungen werden mit bestanden oder nicht bestanden bewertet. Bestanden Studienleistungen sind Voraussetzung zur Teilnahme an einer Modulabschlussprüfung.

**Qualifizierte Teilnahme** Die qualifizierte Teilnahme wird in der Regel durch eine Praktikumsarbeit mit anschließendem Gespräch nachgewiesen. Eine qualifizierte Teilnahme liegt dann vor, wenn die erbrachten Leistungen erkennen lassen, dass eine mehr als nur oberflächliche Beschäftigung mit den Gegenständen, die einer Aufgabenstellung zugrunde lagen, stattgefunden hat. Qualifizierte Teilnahmen sind zum Bestehen eines Moduls nachzuweisen.

**Qualifizierte Teilnahme im Studium Generale** In den Veranstaltungen des Studium Generale, bei denen keine Prüfung im Studium Generale abgelegt wird, ist die qualifizierte Teilnahme nachzuweisen.

## 1.4 Liste der nichtkognitive Kompetenzen

Dieser Studiengang baut die folgenden nichtkognitive Kompetenzen auf:

### **Einsatz und Engagement**

- Gefühl der Verpflichtung informatorische Aufträge zu erfüllen
- Durchhaltevermögen bei der Bearbeitung informatischer Aufträge

### **Empathie**

- Fähigkeit zum Perspektiv- und Rollenwechsel
- Fähigkeit sich in informatikfremde Personen hineinzuversetzen
- Erkennen der Anliegen informatikfremder Personen

### **Gruppenarbeit**

Die Fähigkeit, effektiv und effizient in Gruppen bis zu mittlerer Größe (ca. 15 Personen) zu arbeiten.

### **Haltung und Einstellung**

- Affinität gegenüber informatischen Problemen
- Bereitschaft sich informatischen Herausforderungen zu stellen
- Sozial-kommunikative Fähigkeiten als bedeutsam beurteilen

### **Kooperationskompetenz**

- Hilfs- und Kooperationsbereitschaft
- Sprachkompetenz
- Kommunikative Fähigkeiten
- Diskussionsbereitschaft gegenüber informatischen Themen
- Informatische Themen präsentieren können
- Fähigkeit und Bereitschaft informatisches Wissen weiterzugeben
- Fähigkeit und Bereitschaft zu konstruktiver Kritik
- Fähigkeit und Bereitschaft Absprachen zu treffen und einzuhalten
- Bereitschaft entlang der Absprachen zu handeln
- Bereitschaft fremde Ideen anzunehmen

### **Lernkompetenz**

- Fähigkeit und Bereitschaft zu lebenslangem Lernen
- Fähigkeit und Bereitschaft zu problemorientiertem Lernen
- Fähigkeit und Bereitschaft kooperativem Lernen
- Fähigkeit zur Selbstorganisation von Lernprozessen und zu selbstständigem Lernen

### **Lernmotivation**

- Bereitschaft informatische Fähigkeiten und informatorisches Wissen zu erweitern
- Bereitschaft informatische Aufträge zu erfüllen

### **Medienkompetenz**

- Nutzung problemorientierter Lern- und Entwicklungsumgebungen
- Nutzung von Werkzeugen zum wissenschaftlichen Schreiben
- Nutzung von Werkzeugen zum Präsentieren wissenschaftlicher Resultate

### **Motivationale und volitionale Fähigkeiten**

- Offenheit neuen Ideen und Anforderungen gegenüber
- Bereitschaft neue und unvertraute Lösungswege anzuwenden
- Kritikfähigkeit gegenüber einem und reflektierten Umgang mit rezeptartigen Lösungswegen

### **Schreib- und Lesekompetenz (wissenschaftlich)**

- Fähigkeit Quellen zu recherchieren und reflektiert zu beurteilen
- Fähigkeit informatische Sachverhalte sinnvoll zu strukturieren
- Fähigkeit eigene Ideen von anderen korrekt abzugrenzen (Vermeidung von Plagiaten)

### **Selbststeuerungskompetenz**

- Verbindlichkeit
- Disziplin
- Termintreue
- Kompromissbereitschaft
- Übernahme von Verantwortung
- Geduld
- Selbstkontrolle
- Gewissenhaftigkeit
- Zielorientierung
- Motivation
- Aufmerksamkeit

## Kapitel 2

# Studienrichtungen



## 2.1 Algorithm Design

Studienrichtung	Algorithm Design
Koordination	Prof. Dr. rer. nat. Johannes Blömer Codes und Kryptographie Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Advanced Algorithms (S. 21)</li> <li>• Advanced Complexity Theory (S. 25)</li> <li>• Advanced Distributed Algorithms and Data Structures (S. 30)</li> <li>• Algorithmic Game Theory (S. 34)</li> <li>• Algorithms for Highly Complex Virtual Scenes (S. 36)</li> <li>• Clustering Algorithms (S. 50)</li> <li>• Computational Geometry (S. 54)</li> <li>• Foundations of Cryptography (S. 63)</li> <li>• Linear and Integer Optimization (S. 84)</li> <li>• Public-Key Cryptography (S. 116)</li> <li>• Routing and Data Management in Networks (S. 120)</li> </ul>
Beschreibung	
<p>In diesem Vertiefungsgebiet konzentrieren sich Studierende auf</p> <ul style="list-style-type: none"> <li>• wesentliche Techniken zum Entwurf effizienter Algorithmen</li> <li>• Anwendungsgebiete des Entwurfs effizienter Algorithmen, z.B. Computergrafik, Netzwerke, Big Data, ...</li> <li>• Grenzen für den Entwurf effizienter Algorithmen, d.h. Komplexitätstheorie</li> <li>• konstruktiver Anwendungen der Grenzen des Entwurfs effizienter Algorithmen in Kryptographie und IT-Sicherheit</li> </ul>	

## 2.2 Computer Systems

Studienrichtung	Computer Systems
Koordination	Prof. Dr. Marco Platzner Technische Informatik Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Adaptive Hardware and Systems (S. 18)</li> <li>• Advanced Computer Architecture (S. 28)</li> <li>• Algorithms for Synthesis and Optimization of Integrated Circuits (S. 39)</li> <li>• Architektur paralleler Rechnersysteme (S. 41)</li> <li>• Empiric performance evaluation (S. 61)</li> <li>• Hardware/Software Codesign (S. 70)</li> <li>• High-Performance Computing (S. 72)</li> <li>• Intelligence in Embedded Systems (S. 74)</li> <li>• Reconfigurable Computing (S. 118)</li> <li>• VLSI Testing (S. 140)</li> </ul>
Beschreibung	
<p>Das Vertiefungsgebiet "Computersysteme" behandelt vertiefend und im technischen Detail verschiedene Aspekte von modernen Computersystemen. Im Vordergrund stehen dabei die Analyse und Bewertung von Rechnerarchitekturen, systematische Methoden für den Entwurf und die Optimierung von Computersystemen, insbesondere das Zusammenspiel von Hardware und Software, sowie Programmiermodelle und -methoden für die stark an Bedeutung gewinnenden parallelen und spezialisierten Rechnerarchitekturen.</p>	

## 2.3 Intelligence and Data

Studienrichtung	Intelligence and Data
Koordination	Prof. Dr. Eyke Hüllermeier Intelligente Systeme Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Clustering Algorithms (S. 50)</li> <li>• Intelligence in Embedded Systems (S. 74)</li> <li>• Interactive Data Visualization (S. 76)</li> <li>• Logic Programming for Artificial Intelligence (S. 89)</li> <li>• Logic and Automated Reasoning (S. 86)</li> <li>• Machine Learning I (S. 92)</li> <li>• Machine Learning II (S. 94)</li> <li>• Planning and Heuristic Search (S. 110)</li> </ul>
Beschreibung	
	<p>Intelligente Systeme sind Computersysteme, deren Verhalten durch Methoden und Algorithmen der Künstlichen Intelligenz (KI) gesteuert wird. Solche Systeme gewinnen kontinuierlich an Bedeutung, nicht nur auf wissenschaftlicher Ebene innerhalb der Informatik, sondern auch im sozialen und gesellschaftlichen Kontext: Autonome oder teilautonome Systeme wie Serviceroboter, selbstfahrende PKWs oder medizinische Diagnosesysteme werden unser privates und berufliches Leben in absehbarer Zukunft tiefgreifend verändern. Neben methodischen Fortschritten und einer Steigerung der Rechenleistung durch schnellere Hardware ist die rasante Entwicklung von KI-Systemen in der letzten Dekade vor allem einer Datenexplosion zu verdanken: Die Verfügbarkeit großer Mengen von Daten oder sensorisch erfasster Beobachtungen aus ihrer Umgebung versetzt intelligente Systeme in die Lage, ihr Verhalten durch Adaption und Lernen selbständige zu optimieren.</p> <p>Dieses Modul greift wichtige Aspekte des Entwurfs intelligenter Systeme auf und vermittelt entsprechende theoretische und methodische Grundlagen. Die Inhalte des Moduls erstrecken sich von Themen wie Maschinelles Lernen und Datenanalyse über das Datenmanagement bis hin zur graphischen Datenverarbeitung und Anwendungen in der Robotik und Schwarmintelligenz.</p>

## 2.4 Networks and Communication

Studienrichtung	Networks and Communication
Koordination	Prof. Dr. rer. nat. Holger Karl Rechnernetze Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Advanced Distributed Algorithms and Data Structures (S. 30)</li> <li>• Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 44)</li> <li>• Empiric performance evaluation (S. 61)</li> <li>• Future Internet (S. 68)</li> <li>• Mobile Communication (S. 99)</li> <li>• Network Simulation (S. 106)</li> <li>• Networked Embedded Systems (S. 108)</li> <li>• Routing and Data Management in Networks (S. 120)</li> <li>• Vehicular Networking (S. 138)</li> </ul>
Beschreibung	<p>Das Vertiefungsgebiet “Netze und Kommunikation” lehrt Architekturen, Methoden und Systeme moderner Kommunikationstechnik. Hierzu werden Methoden unterschiedlicher Abstraktionsebenen untersucht, beginnend bei der physikalischen Übertragung bis hin zum Anwendungsentwurf in verteilten Umgebungen. Dabei werden unterschiedliche Systemklassen behandelt, von klassischer Mobilkommunikation über ad hoc Netze und Fahrzeugkommunikation bis zur Vernetzung in Rechenzentren und Architekturen des zukünftigen Internets. Dabei wird die Brücke zu verteilten System hergestellt. Neben Fragen des Architekturentwurfs, der Methoden- und Protokollgestaltung steht dabei stets die Frage der Bewertung solcher Verfahren im Raum; hierzu werden experimentelle und statistische Verfahren besprochen.</p>

## 2.5 Software Engineering

Studienrichtung	Software Engineering
Koordination	Prof. Dr. Gregor Engels Datenbank- und Informationssysteme Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Advanced Compiler Construction (S. 23)</li> <li>• Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 32)</li> <li>• Build It, Break It, Fix It (S. 47)</li> <li>• Compiler Construction (S. 52)</li> <li>• Deductive Verification (S. 56)</li> <li>• Designing code analyses for large-scale software systems (S. 58)</li> <li>• Empiric performance evaluation (S. 61)</li> <li>• Fundamentals of Model-Driven Engineering (S. 65)</li> <li>• High-Performance Computing (S. 72)</li> <li>• Kontextuelle Informatik (S. 79)</li> <li>• Language-Based Security (S. 82)</li> <li>• Logic Programming for Artificial Intelligence (S. 89)</li> <li>• Model Checking (S. 101)</li> <li>• Model-Based Interface Development (S. 103)</li> <li>• Software Analysis (S. 126)</li> <li>• Software Quality Assurance (S. 128)</li> <li>• Type Systems for Correctness and Security (S. 132)</li> <li>• Usability Engineering Practice (S. 135)</li> </ul>

**Beschreibung**

In dieser Studienrichtung können die Studierenden Konzepte, Sprachen, Methoden, Techniken und Werkzeuge für eine systematische Entwicklung von Softwaresysteme erlernen. Dies umfasst

- konstruktive Techniken zur Realisierung von funktionalen und nicht-funktionalen Aspekten eines Softwaresystems,
- formale and informelle analytische Techniken, um eine hohe Qualität eines Systems zu erzielen und
- systematische Techniken, um situationspezifische Vorgehensmodelle zu definieren.

# Kapitel 3

# Module

### 3.1 Wahlpflichtmodul: Adaptive Hardware and Systems

Modulname	Adaptive Hardware and Systems / Adaptive Hardware and Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Adaptive Hardware and Systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Adaptive Hardware and Systems: Vorlesung ( 45h / 105h / EN / SS / 25 ) Adaptive Hardware and Systems: Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Adaptive Hardware and Systems: Grundlegende Kenntnisse in Digitaltechnik, Rechnerarchitektur, Programmierkenntnisse	
<b>Inhalte</b>	
<p>Adaptive Hardware and Systems: Adaptation bezeichnet die Fähigkeit eines Systems, die eigene Funktions- und Leistungsfähigkeit an sich verändernde Bedingungen anzupassen und aufrechtzuerhalten. Die Vorlesung konzentriert sich auf adaptive Hardwaresysteme. Nach einer kurzen Einführung rekonfigurierbarer Bausteine, behandelt die Vorlesung die Prinzipien der Computational Intelligence für die Umsetzung der Adaptations- und Optimierungsmechanismen.</p> <p>In Laborübungen werden mitunter folgende Programmieraufgaben gelöst: lernende Systeme (adaptive Signalklassifikation), Optimierung Chipentwurf (Floorplanning, Placement) und parallele Optimierung von Prozessorcaches (PC2 Cluster).</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Nach Abschluss des Moduls können die Studierenden</p> <ul style="list-style-type: none"> <li>• die Prinzipien moderner Metaheuristiken erläutern und anwenden,</li> <li>• Optimierungsaufgaben typisieren und formal modellieren,</li> <li>• Zielfunktionen und Randbedingungen definieren, Lösungsansätze entwickeln und</li> <li>• wesentliche Herausforderungen beim automatisierten Entwurf, Optimierung und Adaption digitaler Schaltungen und moderner Computersysteme benennen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	



<b>Methodische Umsetzung</b>
Adaptive Hardware and Systems: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Selbststudium und Diskussion wissenschaftlicher Publikationen</li> <li>• Praktische Programmierprojekte</li> <li>• Lösung paralleler Optimierungsaufgaben am PC2 Cluster</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Paul Kaufmann
<b>Lernmaterialien, Literaturangaben</b>
Adaptive Hardware and Systems: Folien; ausgewählte Fachartikel; Lehrbücher <ul style="list-style-type: none"> <li>• Weicker, Karsten, "Evolutionäre Algorithmen", Springer, 2007. ISBN 978-3-8351-9203-4</li> <li>• Kruse et al.: "Computational Intelligence - A Methodological Introduction", Springer, 2013. ISBN 978-1-4471-5012-1</li> <li>• Kruse et al.: "Computational Intelligence [DE]", Vieweg+Teubner-Verlag, Wiesbaden, 2011. ISBN 978-3-8348-1275-9</li> <li>• Wang et al.: "Électronic Design Automation", Morgan Kaufmann, 2009. ISBN: 0-1237-4364-8</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.2 Wahlpflichtmodul: Advanced Algorithms

Modulname	Advanced Algorithms / Advanced Algorithms
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Advanced Algorithms : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Algorithms: Vorlesung ( 45h / 105h / EN / WS / 50 ) Advanced Algorithms: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Algorithms: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen werden vorausgesetzt.	
<b>Inhalte</b>	
Advanced Algorithms: Dieser Kurs präsentiert fortgeschrittenen Algorithmen und algorithmische Paradigmen für grundlegenden Probleme. Insbesondere werden dabei Methoden wie Randomisierung und Derandomisierung, sowie die Konzepte von Approximations- und Onlinealgorithmen anhand wichtiger algorithmischer Probleme vorgestellt. In allen Fällen werden Korrektheitsbeweise und Laufzeitanalysen durchgeführt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden wenden fortgeschrittenen algorithmische Entwurfsmethoden wie Randomisierung, Approximation und Onlinealgorithmen auf neue Probleme und analysieren sie unter Nutzung von kombinatorischen und wahrscheinlichkeitstheoretischen Methoden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Algorithms: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb.</li> <li>• Übungen in Kleingruppen.</li> <li>• erwartete Aktivitäten der Studierenden: aktive Mitarbeit bei Präsenzübungen, Hausaufgaben.</li> </ul>	

<ul style="list-style-type: none"> <li>• Übungsblätter, Lösungen werden in Übungsgruppen vorgestellt und diskutiert.</li> <li>• In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Friedhelm Meyer auf der Heide
<b>Lernmaterialien, Literaturangaben</b>
Advanced Algorithms: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter
<b>Sonstige Hinweise</b>
keine

### 3.3 Wahlpflichtmodul: Advanced Compiler Construction

Modulname	Advanced Compiler Construction / Advanced Compiler Construction
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Compiler Construction : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Compiler Construction: Vorlesung ( 45h / 105h / EN / WS / 25 ) Advanced Compiler Construction: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Compiler Construction: Grundlegende Kenntnisse aus dem Compilerbau werden vorausgesetzt.	
<b>Inhalte</b>	
Advanced Compiler Construction: Ziel des Kurses ist es, fortgeschrittene Techniken des Compilerbaus (u.a. auch aus dem Security Bereich, siehe LVA "Sprach-basierte Sicherheit") zu vermitteln.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Den Studenten erwerben fundierte Kenntnisse aus dem Bereich der Just-in-Time Compiler. Probleme dynamischer Programmiersprachen werden eingehend erläutert und verschiedene Lösungen werden anhand konkreter Implementierungsbeispiele veranschaulicht.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Lernkompetenz</li> <li>Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Compiler Construction: <ul style="list-style-type: none"> <li>Vorlesung mit Beamer und Tafelanschrieb.</li> <li>Begleitende Implementierung als Übungsteil.</li> </ul>	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof.Dr. Stefan Brunthaler
<b>Lernmaterialien, Literaturangaben</b>
Advanced Compiler Construction: keine
<b>Sonstige Hinweise</b>
keine

### 3.4 Wahlpflichtmodul: Advanced Complexity Theory

Modulname	Advanced Complexity Theory / Advanced Complexity Theory
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Complexity Theory : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Complexity Theory: Vorlesung ( 45h / 105h / EN / WS oder SS / 25 ) Advanced Complexity Theory: Übung ( 30h / 0h / EN / WS oder SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Complexity Theory: Grundlagen über Komplexitätstheorie (u.a. Turingmaschinen, NP-Vollständigkeit)	
<b>Inhalte</b>	
<p>Advanced Complexity Theory: Komplexitätstheorie beschäftigt sich mit der Bestimmung der Größe von Ressourcen (z.B. Laufzeit, Speicherverbrauch), die notwendig und hinreichend ist für die Lösung eines bestimmten algorithmischen Problems (z.B. Problem des Handlungsreisenden (TSP)) auf einem bestimmten Computermodell (z.B. Turing-Maschine). Ein Ansatz ist die Komplexitätsklassen wie z.B. P, NP, PSPACE zu definieren, um die Problemkomplexität mit Hilfe der Vollständigkeit in einer solchen Klasse zu klassifizieren, wie z.B. die berühmte Klasse der NP-vollständigen Probleme. Dies ergibt bedingte Aussagen wie „Wenn NP nicht gleich P, dann ist TSP nicht in Polynomialzeit lösbar“. Dieser Zweig der Komplexitätstheorie wird oft als strukturelle Komplexitätstheorie bezeichnet. Im Gegensatz dazu ist das Beweisen expliziter Untergrenzen für bestimmte Probleme das Thema der so genannten konkreten Komplexitätstheorie. Da niemand derzeit in der Lage ist superlineare Zeitschranken für explizit definierte Probleme in allgemeinen Rechenmodellen wie Turingmaschinen zu beweisen, betrachtet man etwas eingeschränkt Modelle wie 1-Band Turingmaschinen, monotone Bool'sche Schaltkreise, Bool'sche Schaltkreise mit beschränkter Tiefe, algebraische Berechnungsmodelle und verschiedene Arten von parallelen Berechnungsmodellen. Die Vorlesung gibt eine Übersicht von Ansätzen um solche unteren Schranke in verschiedenen Modellen zu beweisen.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Die Studierenden beherrschen die wichtigsten Techniken im Bereich der Komplexitätstheorie wie Reduktionen, Diagonalisierung, Randomisierung und Relativierung. Sie können entscheiden, in welche Komplexitätsklassen sich der Speicherplatzbedarf und die Laufzeitanforderungen von algorithmischen Problemen einordnen lassen. Sie können mit Hilfe der Konzepte der Komplexitätstheorie Hypothesen aufstellen und diese falsifizieren oder verifizieren. Sie können die Zusammenhänge zwischen Komplex-</p>	

itätsklassen wie P und NL erläutern.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Advanced Complexity Theory:</p> <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Mündliche Prüfung (ca. 40 Minuten)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: schriftliche Übungsaufgaben  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Friedhelm Meyer auf der Heide
<b>Lernmaterialien, Literaturangaben</b>
<p>Advanced Complexity Theory: C.H. Papadimitriou, Computational Complexity, Addison-Wesley, S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press; Foliensatz der Vorlesung, Übungsblätter</p>



<b>Sonstige Hinweise</b>
keine

### 3.5 Wahlpflichtmodul: Advanced Computer Architecture

Modulname	Advanced Computer Architecture / Advanced Computer Architecture
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Computer Architecture : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Computer Architecture: Vorlesung ( 45h / 105h / EN / WS / 50 ) Advanced Computer Architecture: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Computer Architecture: Grundlegende Kenntnisse in Rechnerarchitektur.	
<b>Inhalte</b>	
Advanced Computer Architecture: Die Lehrveranstaltung vermittelt die wesentliche Konzepte und Methoden, die beim Entwurf moderner Prozessoren Verwendung finden. Es werden Ansätze zur Nutzung von Parallelität auf der Instruktions-, Daten- und Thread-Ebene besprochen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>die Prinzipien modernen Speicherhierarchien zu erklären,</li> <li>die verschiedenen Ebenen der Parallelität zu analysieren,</li> <li>die Eignung unterschiedlicher Architekturkonzepte einzuschätzen und dadurch</li> <li>moderne Entwicklungen der Rechnerarchitektur zu bewerten.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Gruppenarbeit</li> <li>Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Computer Architecture: <ul style="list-style-type: none"> <li>Vorlesung mit Beamer und Tafel</li> <li>Interaktive Übungen im Hörsaal</li> <li>Rechnerübungen mit Simulationswerkzeugen</li> <li>Analyse von Fallbeispielen</li> </ul>	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Advanced Computer Architecture: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• Hennessey, Patterson: Computer Architecture: A Quantitative Approach (5th edition), Morgan Kaufmann, 2012.</li> <li>• Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.6 Wahlpflichtmodul: Advanced Distributed Algorithms and Data Structures

Modulname	Advanced Distributed Algorithms and Data Structures / Advanced Distributed Algorithms and Data Structures
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Advanced Distributed Algorithms and Data Structures : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Distributed Algorithms and Data Structures: Vorlesung ( 45h / 105h / EN / WS / 30 ) Advanced Distributed Algorithms and Data Structures: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Distributed Algorithms and Data Structures: Algorithmen und Datenstrukturen, verteilte Algorithmen und Datenstrukturen	
<b>Inhalte</b>	
Advanced Distributed Algorithms and Data Structures: Die Vorlesung stellt fortgeschrittene Methoden vor, um hochskalierbare verteilte Algorithmen und Datenstrukturen zu entwickeln. Die Vorlesung teilt sich dabei in separate Bereiche auf, die aktuell relevant für den Bereich der verteilten Systeme sind. Dazu gehören lokalitätserhaltende Systeme, robust Informationssysteme, und programmierbare Materie.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende lernen fortgeschrittene Methoden und Verfahren für aktuell sehr relevante verteilte Systeme kennen. Sie können Verfahren an neue Situationen anpassen und deren Komplexität bestimmen. Sie können grundlegende Verfahren implementieren.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Advanced Distributed Algorithms and Data Structures: Vorlesung mit Übungen und Softwareprojekt	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Christian Scheideler
<b>Lernmaterialien, Literaturangaben</b>
Advanced Distributed Algorithms and Data Structures: Skript
<b>Sonstige Hinweise</b>
keine

### 3.7 Wahlpflichtmodul: Advanced Software Engineering: Methods, Architectures, Industrial Applications

Modulname	Advanced Software Engineering: Methods, Architectures, Industrial Applications / Advanced Software Engineering: Methods, Architectures, Industrial Applications
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Advanced Software Engineering: Methods, Architectures, Industrial Applications : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Vorlesung ( 45h / 105h / EN / WS / 92 ) Advanced Software Engineering: Methods, Architectures, Industrial Applications: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Modellbasierte Softwareentwicklung	
<b>Inhalte</b>	
Advanced Software Engineering: Methods, Architectures, Industrial Applications: In der Veranstaltung werden fortgeschrittene Konzepte und Methoden der Softwaretechnik vermittelt. Hierzu gehören Ansätze zur Entwicklung von Architekturen, moderne Softwareentwicklungsmethoden sowie Elemente des Projektmanagements. Im Einzelnen werden die folgenden Themen behandelt: - Architekturen (serviceorientierte Architekturen, Microservices, Web Services, Architekturen für mobile Anwendungen, Architekturframeworks, Quasar Enterprise) - Methoden (Agile Entwicklungsmethoden wie Scrum, Situational Method Engineering, Requirements Engineering, DevOps) - Projektmanagement (Aufwandsschätzung, Ökonomie von Softwareprojekten) Die Veranstaltung beinhaltet auch Vorträge von erfahrenen Experten aus der Industrie, die über den Einsatz der modernen Software-Engineering-Techniken in der Praxis berichten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage, wesentliche Konzepte einer modernen Softwareentwicklung zu benennen und einzusetzen. Sie kennen die wesentlichen Techniken, um Architekturen von Softwaresystemen zu entwickeln, Softwareentwicklungsmethoden situationsgerecht anzupassen und einzusetzen sowie Pro-	

jektmanagementaufgaben wie z.B. Aufwandsschätzung durchzuführen.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Empathie</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. In praktischen Übungen werden die erlernten Kenntnisse angewendet.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gregor Engels
<b>Lernmaterialien, Literaturangaben</b>
Advanced Software Engineering: Methods, Architectures, Industrial Applications: Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.8 Wahlpflichtmodul: Algorithmic Game Theory

Modulname	Algorithmic Game Theory / Algorithmic Game Theory
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Algorithmic Game Theory : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithmic Game Theory: Vorlesung ( 45h / 105h / EN / WS oder SS / 25 ) Algorithmic Game Theory: Übung ( 30h / 0h / EN / WS oder SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithmic Game Theory: Algorithmen und Datenstrukturen, Lineare Algebra, Wahrscheinlichkeitstheorie	
<b>Inhalte</b>	
Algorithmic Game Theory: In dieser Vorlesung werden die wichtigsten Ergebnisse und Methoden aus dem Gebiet der Algorithmischen Spieltheorie vorgestellt. Es werden zahlreiche Fragestellungen der nicht kooperativen Spieltheorie sowie aus dem Bereich des Social Choice behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende lernen die wesentlichen Konzepte der kooperativen und nicht-kooperativen Spieltheorie kennen, insbesondere Standard Modelle und Lösungskonzepte. Sie verstehen, eine Vielzahl von Algorithmischentechniken und Komplexitätsresultate zur Berechnung spieltheoretische Lösungskonzepte. Sie können Lösungskonzepte, Algorithmen und Komplexitätsresultate auf neue Spiele, die Varianten bekannter Beispiele sind, anwenden. Sie verstehen den Stand der algorithmischen Forschung in einigen Bereichen, einschließlich neuer Trends und offene Probleme.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Algorithmic Game Theory:	



<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Jun.-Prof. Dr. Alexander Skopalik
<b>Lernmaterialien, Literaturangaben</b>
Algorithmic Game Theory: Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. 2007. Algorithmic Game Theory. Cambridge University Press, New York, NY, USA. Slides Exercises
<b>Sonstige Hinweise</b>
keine

### 3.9 Wahlpflichtmodul: Algorithms for Highly Complex Virtual Scenes

Modulname	Algorithms for Highly Complex Virtual Scenes / Algorithms for Highly Complex Virtual Scenes
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Algorithmen für hochkomplexe virtuelle Szenen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithmen für hochkomplexe virtuelle Szenen: Vorlesung ( 45h / 105h / EN / WS / 40 ) Algorithmen für hochkomplexe virtuelle Szenen: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithmen für hochkomplexe virtuelle Szenen: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen werden angenommen.	
<b>Inhalte</b>	
Algorithmen für hochkomplexe virtuelle Szenen: Walkthrough-Systeme erlauben das Betrachten und Durchlaufen von virtuellen 3D-Szenen und finden Anwendung in Architekturprogrammen, Simulationen, oder Spielen. Die Effizienz von Echtzeit-Rendering Algorithmen ist entscheidend für eine flüssige und schnelle Darstellung der virtuellen 3D-Szenen in einem Walkthrough-System. Es gibt verschiedene algorithmische Ansätze, um hochkomplexe geometrische 3D-Daten zu reduzieren und eine Darstellung der Daten in Echtzeit zu erreichen. In der Vorlesung werden algorithmische Ansätze aus den Bereichen Visibility-Culling, Simplification, Level of Detail, Image-Based Rendering und weitere vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden können die wichtigsten Techniken im Bereich der Echtzeit-Darstellung virtueller 3D-Szenen anwenden. Sie können entscheiden, in welcher virtuellen 3D-Szene welcher Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
<p>Algorithmen für hochkomplexe virtuelle Szenen:</p> <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben</li> <li>• Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt</li> <li>• In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Mündliche Prüfung (ca. 40 Minuten)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: schriftliche Übungsaufgaben  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Matthias Fischer
<b>Lernmaterialien, Literaturangaben</b>
<p>Algorithmen für hochkomplexe virtuelle Szenen: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter</p> <ul style="list-style-type: none"> <li>• Real-Time Rendering; Tomas Akenine-Möller, Eric Haines; AK Peters, 2002.</li> <li>• Level of Detail for 3D Graphics; David Luebke, Martin Reddy, Jonathan D. Cohen; Morgan Kaufmann Publishers, 2002.</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.10 Wahlpflichtmodul: Algorithms for Synthesis and Optimization of Integrated Circuits

Modulname	Algorithms for Synthesis and Optimization of Integrated Circuits / Algorithms for Synthesis and Optimization of Integrated Circuits
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Algorithms for Synthesis and Optimization of Integrated Circuits : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithms for Synthesis and Optimization of Integrated Circuits: Vorlesung ( 45h / 105h / EN / SS / 30 )	
Algorithms for Synthesis and Optimization of Integrated Circuits: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithms for Synthesis and Optimization of Integrated Circuits: Kenntnisse aus Digitaltechnik sind hilfreich.	
<b>Inhalte</b>	
Algorithms for Synthesis and Optimization of Integrated Circuits: Die Veranstaltung behandelt die wesentlichen Schritte bei der Synthese digitaler Schaltungen und geht speziell auf die Übersetzung von Beschreibungen in Hardwarebeschreibungssprachen in Schaltungen ein. Weiterhin werden die wichtigsten Techniken für die Logikoptimierung diskutiert. In praktischen Übungen wird die effiziente Verwendung von Entwurfswerkzeugen geübt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>zwischen den verfügbaren Optimierungsmethoden für den digitalen Schaltungsentwurf auszuwählen,</li> <li>die wesentlichen Probleme bei Entwurf integrierter Schaltungen zu identifizieren und die Tradeoffs beim Schaltungsentwurf zu erkennen, und</li> <li>aktuelle Werkzeuge für den digitalen Schaltungsentwurf zu bewerten.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>Gruppenarbeit</li> <li>Lernkompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
Algorithms for Synthesis and Optimization of Integrated Circuits: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Rechnerübungen mit Hardwaresynthesewerkzeugen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Hassan Ghasemzadeh Mohammadi
<b>Lernmaterialien, Literaturangaben</b>
Algorithms for Synthesis and Optimization of Integrated Circuits: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• Micheli, Giovanni De. Synthesis and optimization of digital circuits. McGraw-Hill Higher Education, 1994.</li> <li>• Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.11 Wahlpflichtmodul: Architektur paralleler Rechnersysteme

Modulname	Architektur paralleler Rechnersysteme / Architectures of Parallel Computer Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Architektur paralleler Rechnersysteme : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Architektur paralleler Rechnersysteme: Vorlesung ( 45h / 105h / DE / SS / 20 ) Architektur paralleler Rechnersysteme: Übung ( 30h / 0h / DE / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Architektur paralleler Rechnersysteme: Grundlagen der Rechnerarchitektur	
<b>Inhalte</b>	
Architektur paralleler Rechnersysteme: Diese Veranstaltung führt in Rechnerarchitekturen der wichtigsten Parallelrechner und in die Nutzung dieser Systeme ein. Der Schwerpunkt der Veranstaltung liegt auf High-Performance-Computer.	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Studierende benennen und erklären Programmierparadigmen paralleler Programmiersprachen. Sie beherrschen grundlegende Sprachkonstrukte und Bibliotheksfunktionen wichtiger paralleler Programmiersprachen/-umgebungen (z.B. OpenMP, POSIX-Threads, MPI, PGAS) und können deren Einsatzgebiete benennen. Studierende sind in der Lage sowohl einige aktuelle HPC-Systeme als auch moderne Prozessoren mit deren Eigenschaften zu beschreiben. Sie erkennen bedeutende Trends (Power Wall, Memory Wall, ILP Wall) denen diese Systeme unterliegen. Studierende benennen und erklären allgemein genutzte Klassifikation von Parallelrechnern. Sie erklären die wichtigsten Strukturbausteine und Operationsprinzipien paralleler Rechnersysteme. Sie beherrschen die theoretische Beschreibung des Skalierungsverhaltens (Amdahl, Gustafson) und die quantitativen Bewertungen von Parallelrechnern. Studierende benennen und erklären Architekturmerkmale skalierbarer speichergekoppelte Systeme. Sie beherrschen unterschiedliche Techniken zur Aufrechterhaltung der Speicherkonsistenz und -kohärenz in busbasierten Systemen (Invalidierungs-, Update-Protokolle). Sie sind in der Lage Techniken zur Steigerung der Leistungsfähigkeit dieser Systeme zu beschreiben (Multi-Level-Caches, transiente Zustände, Split-Transaktion-Busse). Studierende erklären Mechanismen zur Synchronisation (Locks, Barrieren) in Parallelrechnern. Studierende demonstrieren Kenntnisse in Aufrechterhaltung der Cache-Kohärenz von skalierbaren Rechnersystemen (hierarchisches Snooping, Directories). Sie beherrschen</p>	

<p>Techniken zur Steigerung der Leistungsfähigkeit solcher Systeme (z.B. Latenz-Verbesserung, Durchsatzserhöhung). Studierende beschreiben Verfahren basierend auf Token Coherence. Studierende benennen und erklären grundlegende Eigenschaften von Cluster-Architekturen. Sie können die in dem Bereich eingesetzte Kommunikationsnetzwerke topologisch beschreiben und bewerten (z.B. Grad, Durchmesser, Bisektion). Sie beherrschen Kommunikationstechniken der Hochgeschwindigkeitsnetzwerke (Wormhole Routing, Virtual Cut-Through) und Routing-Verfahren (tabellenbasiertes Routing, Source-Routing). Sie beherrschen Beweistechniken zur Sicherstellung der Deadlock-Freiheit von Routings. Studierende können die Eigenschaften existenter Interconnects (z.B. InfiniBand, OmniPath) benennen. Die Studierenden sind in der Lage, wesentliche Architektureigenschaften von parallelen Rechnersystemen zu erkennen und deren Eignung für bestimmte Anwendungsgebiete festzustellen. Die Kenntnisse können dazu eingesetzt werden um hohe Rechenleistungen auf HPC-Systemen zu erzielen und vorhandene Ressourcen effizient zu nutzen.</p>
<p><b>Nichtkognitive Kompetenzen</b></p>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>
<p><b>Methodische Umsetzung</b></p>
<p>Architektur paralleler Rechnersysteme: Einsatz von Folien. In der Übung wird ein Zugang zu vorhandenen HPC-Systemen genutzt um den praktischen Umgang mit den Rechnern zu üben und die Kenntnisse der Vorlesung zu vertiefen.</p>
<p><b>Prüfungsleistung (Dauer)</b></p>
<p>Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<p><b>Moduleilprüfungen</b></p>
<p>keine</p>
<p><b>Studienleistung / qualifizierte Teilnahme</b></p>
<p>Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<p><b>Voraussetzungen für die Teilnahme an der Prüfung</b></p>
<p>Bestehen der Studienleistung.</p>
<p><b>Voraussetzungen für die Vergabe von Credits</b></p>
<p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>
<p><b>Gewichtung für die Gesamtnote</b></p>
<p>Das Modul wird mit 6 Credits gewichtet.</p>
<p><b>Modul wird in folgenden Studiengängen verwendet</b></p>
<p>–</p>



<b>Modulbeauftragte/r</b>
Dr. Jens Simon
<b>Lernmaterialien, Literaturangaben</b>
Architektur paralleler Rechnersysteme: Foliensatz
<b>Sonstige Hinweise</b>
keine

### 3.12 Wahlpflichtmodul: Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies

Modulname	Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies / Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Vorlesung ( 45h / 105h / EN / SS / 30 )	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Grundlegende Kenntnisse der Kryptographie sind empfehlenswert, jedoch keine Voraussetzung.	
<b>Inhalte</b>	
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: Diese Vorlesung vermittelt Techniken zur Konstruktion dezentraler digitaler Währungen wie z.B. Bitcoin, ein Verständnis für die erreichbaren und erreichten Sicherheitseigenschaften, sowie kryptographische Werkzeuge zum Schutz der Privatsphäre in solchen Anwendungen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende kennen die grundlegenden Konzepte, die modernen verteilten kryptographischen Währungen zugrunde liegen und können diese erklären und anwenden. Dies umfasst kryptographische Hashfunktionen, digitale Signaturverfahren, und Blockchains. Sie kennen die nötigen Sicherheitsanforderungen an diese Bausteine und können einschätzen, inwiefern ein gegebenes Verfahren diese Anforderungen erfüllt. Sie kennen die Begriffe Pseudonymität und Anonymität und ihre genaue Bedeutung und Abgrenzung, sind in der Lage zu untersuchen inwiefern eine gegebene Anwendung diese Eigenschaften besitzt, kennen fortgeschrittene kryptographische Techniken um starke Anonymität zu erreichen, wie zum Beispiel Ring-Signaturen, und wissen wie diese Techniken in modernen Anwendungen eingesetzt werden um die Privatheit von Nutzern zu schützen.	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr., Tibor Jager
<b>Lernmaterialien, Literaturangaben</b>
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Tafelanschrieb</li> <li>• Originalliteratur</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.13 Wahlpflichtmodul: Build It, Break It, Fix It

Modulname	Build It, Break It, Fix It / Build It, Break It, Fix It
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Build It, Break It, Fix It : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Build It, Break It, Fix It: Vorlesung ( 45h / 105h / EN / SS / 20 ) Build It, Break It, Fix It: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Bachelor in Informatik (oder gleichwertig)	
<b>Empfohlene Kenntnisse</b>	
Build It, Break It, Fix It: Fortgeschrittenenkenntnisse der Java Programmiersprache und Software-Sicherheitsanforderungen. Best practices im Bereich der sicheren Softwareentwicklung und Erfahrungen mit dem Auffinden und Ausnutzen von Sicherheitslücken in Software sind hilfreich.	
<b>Inhalte</b>	
<p>Build It, Break It, Fix It: Das Ziel dieser Lehrveranstaltung ist die praktische Vermittlung von Grundprinzipien der sicheren Softwareentwicklung. Sie ist inspiriert vom "Break It, Build It, Fix It Wettbewerb von Ruef et al. [1].</p> <p>Die Veranstaltung ist in drei Phasen aufgeteilt, in denen die Teilnehmenden in Gruppen ihre Fertigkeiten in der Entwicklung von Software, Identifizierung von Sicherheitslücken und Behebung derselben unter Beweis stellen und weiterentwickeln.</p> <p>In der "Build ItPhase entwickeln die Gruppen kleine Softwareprojekte nach einer formalen Spezifikation, die auch Sicherheitsanforderungen enthält. In der "Break ItPhase werden die entwickelten Softwareprodukte unter den Gruppen ausgetauscht mit der Zielsetzungen, Schwachstellen in anderen Implementierungen zu finden und auszunutzen. In der "Fix ItPhase behebt jede Gruppe die in ihrer Software gefundenen Schwachstellen.</p> <p>Die Veranstaltung enthält einen theoretischen Teil, in dem grundsätzliche Vorgehensweisen zur sicheren Softwareentwicklung erläutert, sowie verschiedene Arten von Sicherheitslücken vorgestellt und demonstriert werden. Der Fokus dieser Veranstaltung liegt jedoch auf der praktischen Arbeit in den Gruppen. Da das Finden und Beheben von Schwachstellen in Software eine Vielzahl von Fertigkeiten und einiges an Kreativität erfordert, sollten die Studenten ein gleiches Maß an Eigenmotivation und Selbstorganisation mitbringen.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	

Theoretische/Praktische Kenntnisse im Bereich der sichere Softwareentwicklung, Theoretische/Praktische Kenntnisse in der Auffindung und Ausnutzung von Software-Sicherheitslücken Wissen über verbreitete, reale Software-Sicherheitslücken und Möglichkeiten diese auszunutzen.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Build It, Break It, Fix It:
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: Praktikumsarbeit Qualifizierte Teilnahme: Praktikumsarbeit mit anschließendem Gespräch Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eric Bodden
<b>Lernmaterialien, Literaturangaben</b>
Build It, Break It, Fix It: Das Kursmaterial wird auf der KoaLA Seite des Kurses angeboten werden.

<b>Sonstige Hinweise</b>
keine

### 3.14 Wahlpflichtmodul: Clustering Algorithms

Modulname	Clustering Algorithms / Clustering Algorithms
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Clustering Algorithmen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Clustering Algorithmen: Vorlesung ( 45h / 105h / EN / WS oder SS / 25 ) Clustering Algorithmen: Übung ( 30h / 0h / EN / WS oder SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Clustering Algorithmen: Algorithmen und Datenstrukturen, Lineare Algebra, Wahrscheinlichkeitstheorie	
<b>Inhalte</b>	
Clustering Algorithmen: Im Zentrum dieser Vorlesung steht eines der wichtigsten Werkzeuge der Datenanalyse. das Clustering. Clustering ist der Prozess des Aufteilen von Daten in sinnvolle oder nützliche Teilmengen. Eine solche Aufteilung sollte die natürliche Struktur der Daten widerspiegeln. Dies kann z.B. bedeuten, dass jede Teilmenge möglichst viele einander ähnliche Objekte einer Datenmenge enthalten soll. Clustering ist eine natürliche Art Daten zu strukturieren und zu analysiere. Es besitzt viele Anwendung in den Natur- und Ingenieurwissenschaften.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende können die wichtigsten Algorithmentechniken des Clustering benennen und erläutern. Sie können entscheiden, in welcher Situation welcher Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen. Sie kennen verschiedene Modellierungen von Clustering, können diese anwenden und anpassen sowie für die Modellierung geeignete Algorithmen einsetzen und bewerten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Clustering Algorithmen: Vorlesung mit Übungen, Lesegruppen, Kurzvorträge	



<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Clustering Algorithmen: David J.C MacKay, Information Theory, Inference and Learning Algorithms Christopher M. Bishop, Pattern Recognition and Machine Learning Folien der Vorlesung
<b>Sonstige Hinweise</b>
keine

### 3.15 Wahlpflichtmodul: Compiler Construction

Modulname	Compiler Construction / Compiler Construction
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Compilerbau : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Compilerbau: Vorlesung ( 45h / 105h / EN / WS / 25 ) Compilerbau: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Compilerbau: Elementare Kenntnisse aus dem Compilerbau, wie etwa aus der LVA "Programming Languages and Compilers" (PLaC)	
<b>Inhalte</b>	
Compilerbau: Die Vorlesung setzt da an, wo "Programming Languages and Compilers" (PLaC, Bachelor Studium) aufgehoert hat. Thematisch steht somit das "Compiler Backend" im Vordergrund und es werden theoretische Grundlagen wichtiger Optimierungen im Compilerbau vermittelt. Die theoretischen Grundlagen werden dann durch einen Praxis-Teil veranschaulicht, bei dem konkrete Implementierungen im LLVM Compiler Framework diskutiert werden.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studenten erwerben fundierte Kenntnisse zur Theorie und Praxis des Compilerbaus. Der vermittelten Grundlagen aus der Programm-Analyse sowie darauf folgende Transformationen/Optimierungen erlauben es dem Studenten, das Optimierungspotential ihrer eigenen Programme voll auszuschöpfen. Der praktische Teil veranschaulicht konkrete Implementierungsdetails und vertieft daher die theoretischen Grundlagen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Compilerbau: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb.</li> <li>• Präsenzübungen in kleinen Gruppen</li> </ul>	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof.Dr. Stefan Brunthaler
<b>Lernmaterialien, Literaturangaben</b>
Compilerbau: <ul style="list-style-type: none"> <li>• Appel: Modern Compiler Construction in ML.</li> <li>• Cooper, Torczon: Engineering a Compiler.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.16 Wahlpflichtmodul: Computational Geometry

Modulname	Computational Geometry / Computational Geometry
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Algorithmische Geometrie : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Algorithmische Geometrie: Vorlesung ( 45h / 105h / EN / SS / 40 ) Algorithmische Geometrie: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Algorithmische Geometrie: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen wird angenommen.	
<b>Inhalte</b>	
Algorithmische Geometrie: Es werden Algorithmen und Datenstrukturen aus dem Bereich der Algorithmischen Geometrie behandelt. Die Grundelemente und Eingabe sind geometrische Daten (Punkte, Linien, Kreise, Polygone, Körper). Die Probleme werden geometrisch formuliert und dafür wird eine algorithmische Lösung mit Hilfe spezieller geometrische Datenstrukturen gesucht. Die Algorithmen werden theoretisch analysiert. Dazu wird Laufzeit und Speicherplatz bestimmt und die Korrektheit der Algorithmen bewiesen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden können die wichtigsten Techniken im Bereich der Algorithmischen Geometrie anwenden. Sie können entscheiden, für welche geometrisch formulierten Probleme welcher Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Algorithmische Geometrie: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> </ul>	

<ul style="list-style-type: none"> <li>• erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben</li> <li>• Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt</li> <li>• In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Matthias Fischer
<b>Lernmaterialien, Literaturangaben</b>
Algorithmische Geometrie: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter <ul style="list-style-type: none"> <li>• Computational Geometry: Algorithms and Applications Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars, Springer-Verlag, 2008</li> <li>• Algorithmische Geometrie Rolf Klein, Springer-Verlag, 2005</li> <li>• Lectures on Discrete Geomtetry Jiri Matousek, Springer-Verlag, 2002.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.17 Wahlpflichtmodul: Deductive Verification

Modulname	Deductive Verification / Deductive Verification
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Deductive Verification : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Deductive Verification: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Deductive Verification: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Deductive Verification: Aussagen- und Prädikatenlogik. Imperative Programmierung.	
<b>Inhalte</b>	
Deductive Verification: In der Vorlesung werden Techniken zum Nachweis der Korrektheit von Programmen vorgestellt. Im Gegensatz zu der Model Checking Vorlesung basieren diese Techniken auf Beweiskalkülen, nicht auf automatischen Methoden. Die Vorlesung wird zuerst das Konzept der Zusicherungen einführen und am Beispiel von Java Programmen illustrieren. Danach werden Hoare-Kalküle zum Nachweis von Zusicherungen erläutert, und zwar sowohl für sequentielle als auch parallele Programme. Für die Beweiskalküle wird ihre Korrektheit und Vollständigkeit untersucht. Zum Schluss werden Beweiskalküle für temporal-logische Formeln vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden kennen die wesentlichen Prinzipien der Programmverifikation. Sie können für sequentielle wie parallele Programme nachweisen, dass diese korrekt sind. Sie wissen wie Zusicherungen zur Beschreibung der Funktionalität von Programmen genutzt werden können. Sie kennen eine temporale Logik und können diese zur Spezifikation von Eigenschaften einsetzen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	

Deductive Verification: Eine Mischung aus Folien und Tafelanschrieb. Interaktion mit Studierenden über Fragen und Beispielaufgaben. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Deductive Verification: K. Apt, E.-R. Olderog: Verification of sequential and concurrent programs Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.18 Wahlpflichtmodul: Designing code analyses for large-scale software systems

Modulname	Designing code analyses for large-scale software systems / Designing code analyses for large-scale software systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Designing code analyses for large-scale software systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Designing code analyses for large-scale software systems: Vorlesung ( 45h / 105h / EN / WS / 30 ) Designing code analyses for large-scale software systems: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Designing code analyses for large-scale software systems: Der vorherige Besuch der Veranstaltung Software Analysis wird empfohlen ist aber nicht zwingend erforderlich. Ein gutes Verständnis von Java und den Prinzipien objektorientierter Programmierung ist hilfreich.	
<b>Inhalte</b>	
Designing code analyses for large-scale software systems: Statische Codeanalysen dienen dazu, automatisiert Fehler und Schwachstellen im Programmcode aufzufinden. Zu diesem Zwecke suchen sie nach bekannten Fehlermustern. In dieser Vorlesung wird erklärt, wie man solche Codeanalysen entwirft, die inter-prozedural sind, also das komplette Programm betrachten, über die Grenzen einzelner Prozeduren hinweg. Der Entwurf solcher Analysen gestaltet sich deshalb sehr schwierig, weil die Analysen oft Millionen von Programmstatements gleichermaßen präzise aber auch effizient verarbeiten müssen. Es werden außerdem Beispielsanalysen aus dem Bereich der IT-Sicherheit besprochen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Durch den Besuch erlenen Studierende... - wichtige Designentscheidungen beim Entwurf automatisierter Codeanalysen richtig zu treffen - welche Algorithmen für Codeanalysen in welchen Anwendungssituationen am besten geeignet sind - wie man Codeanalysen für reale Probleme aus der IT-Sicherheit entwirft - wie man gängige Begriffliche wie Kontext-, Fluss-, Feld-, und Objekt-Sensitivität korrekt interpretiert - welche Limitierungen statische Codeanalysen aufweisen - welche gängige Codeanalysen für Sicherheitsschwachstellen (OWASP Top 10 etc.) existieren, und wie sich diese mit den vorgestellten Algorithmen umsetzen lassen.	



<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Designing code analyses for large-scale software systems: Vorlesung und Gruppenübungen
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eric Bodden
<b>Lernmaterialien, Literaturangaben</b>
Designing code analyses for large-scale software systems: Empfohlene Literatur: * Thomas Reps, Susan Horwitz, and Mooly Sagiv. 1995. Precise interprocedural dataflow analysis via graph reachability. POPL '95 * Shmuel Sagiv, Thomas W. Reps, and Susan Horwitz. 1995. Precise Interprocedural Dataflow Analysis with Applications to Constant Propagation. TAPSOFT '95 * Akash Lal, Thomas Reps, and Gogul Balakrishnan. 2005. Extended weighted pushdown systems. CAV 2005 * Nomair A. Naeem, Ondrej Lhoták, and Jonathan Rodriguez. 2010. Practical extensions to the IFDS algorithm. CC 2010 * Yannis Smaragdakis, Martin Bravenboer, and Ondrej Lhoták. 2011. Pick your contexts well: understanding object-sensitivity. POPL 2011 * Eric Bodden. 2012. Inter-procedural data-flow analysis with IFDS/IDE and Soot. SOAP 2012 * Rohan Padhye, Uday P. Khedker. Interprocedural Data Flow Analysis in Soot using Value Contexts. SOAP 2013

<b>Sonstige Hinweise</b>
keine

### 3.19 Wahlpflichtmodul: Empiric performance evaluation

Modulname	Empiric performance evaluation / Empiric performance evaluation
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Empirische Leistungsbewertung : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Empirische Leistungsbewertung: Vorlesung ( 45h / 105h / EN / SS / 15 ) Empirische Leistungsbewertung: Übung ( 30h / 0h / EN / SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Empirische Leistungsbewertung: Stochastik auf dem Niveau der Bachelor-Ausbildung.	
<b>Inhalte</b>	
Empirische Leistungsbewertung: Die Vorlesung beschreibt Methoden und Verfahren, um experimentelle und simulationsbasierte Leistungsbewertung durchzuführen und statistisch korrekt auszuwerten. Die erlernten Verfahren sind auf eine weite Klasse von Systemen anwendbar.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer können bestimmen, ob ein gegebenes System/Modell einer bestimmten Leistungsbewertungsmethode zugänglich ist. Sie können ein Experiment oder eine Simulation entwerfen und durchführen, die geeigneten stochastischen Modelle auswählen und die Ergebnisse korrekt interpretieren. Sie können statistisch gerechtfertigte Schlüsse ziehen, z.B. ob eines von mehreren System als das beste System angesehen werden kann.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Empirische Leistungsbewertung: Vorlesung mit Folien und Tafelanschrieb; Übungsblätter.	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Empirische Leistungsbewertung: Foliensatz, Übungsblätter, Lehrbuch Kelton & Law, Simulation Modelling and Analysis.
<b>Sonstige Hinweise</b>
keine

### 3.20 Wahlpflichtmodul: Foundations of Cryptography

Modulname	Foundations of Cryptography / Foundations of Cryptography
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Foundations of Cryptography : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Foundations of Cryptography: Vorlesung ( 45h / 105h / EN / SS / 25 ) Foundations of Cryptography: Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Foundations of Cryptography: Basiskenntnisse in IT-Sicherheit und Kryptographie nützlich aber nicht notwendig, Grundkonzepte der Komplexitätstheorie und Wahrscheinlichkeitstheorie	
<b>Inhalte</b>	
Foundations of Cryptography: Wichtige Basiskonzepte moderner Kryptographie werden vorgestellt. Hierzu gehören Verschlüsselungsverfahren, digitale Signaturen, Identifikationsprotokolle und Mehrparteienberechnungen werden vorgestellt. In allen Fällen werden formale Sicherheitsdefinitionen vorgestellt und, ausgehend von mathematisch präzisen Annahmen, beweisbar sichere Konstruktionen entwickelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende verstehen wesentliche Konzepte und Methoden moderner Kryptographie. Sie können für für Sicherheitsprobleme geeignete kryptographische Techniken auswählen. Sie können Basistechniken der Kryptographie kombinieren und modifizieren, neue Sicherheitskonzepte definieren und die Sicherheit der Konstruktionen bezüglich dieses Definitionen beweisen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Foundations of Cryptography: Vorlesung mit Übungen, Lesegruppen	

<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Foundations of Cryptography: Oded Goldreich, Foundations of Cryptography I,II, Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography Folien der Vorlesung
<b>Sonstige Hinweise</b>
keine

### 3.21 Wahlpflichtmodul: Fundamentals of Model-Driven Engineering

Modulname	Fundamentals of Model-Driven Engineering / Fundamentals of Model-Driven Engineering
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Fundamentals of Model-Driven Engineering : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Fundamentals of Model-Driven Engineering: Vorlesung ( 45h / 105h / EN / WS / 30 ) Fundamentals of Model-Driven Engineering: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Fundamentals of Model-Driven Engineering:	
<b>Inhalte</b>	
<p>Fundamentals of Model-Driven Engineering: Die Verwendung von Modellen als primäre Artefakte im Softwareentwicklungsprozess ist eines der Hauptziele der modellgetriebenen Softwareentwicklung (Englisch: Model-Driven Engineering (MDE)). Um dieses Ziel zu erreichen müssen einige Aufgaben gut unterstützt werden: Die Spezifikation neuer Modellierungssprachen (Metamodellierung), die Erzeugung sowie Manipulation von Modellen (Modelltransformationen), und die Möglichkeit Änderungen an einem Modell auf andere betroffene Modellen zu propagieren (Modellsynchronisierung).</p> <p>Nicht nur ein intuitives Verständnis für diese zentralen Konzepte ist wichtig, sondern auch eine präzise Formalisierung wenigstens der wichtigsten Grundkonzepte. Dies ist vor allem dann unabdingbar, wenn man verlässliche Werkzeugunterstützung bieten möchte. In der Vorlesung werden daher Grundkonzepte des MDE eingeführt und mit einfachen Mitteln der Kategorientheorie formalisiert.</p> <p>Die Vorlesung ist mit dem Ziel entworfen besonders Informatikern zugänglich zu sein. Dies wird erreicht durch eine konstruktive und direkte Abbildung aller Definitionen und Beweiskonstruktionen auf lauffähige Programme in Java.</p> <p>Die Vorlesung ergänzt die MDS-D-Vorlesung so, dass Studierende beide Vorlesungen in beliebiger Reihenfolge besuchen können.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende sind in der Lage, selbständig weiterführende Bücher und Papiere über Graphtransformationen zu lesen und neue Definitionen sowie Konstruktionsverfahren zu verstehen. Studierende können die Methoden zur Formalisierung aus der Vorlesung anwenden um neue Strukturen und die regelbasierte Ma-	

nipulation dieser Strukturen analog zu formalisieren. Studierende verstehen und schätzen die Vorteile einer präzisen und konstruktiven Formalisierung von Pattern-Matching, Constraints, und Regeln im Kontext der modellgetriebenen Softwareentwicklung. Studierende kennen einige Grundkonzepte der Kategorientheorie und verstehen wie und wieso diese für die Formalisierung von Graphtransformationen verwendet werden.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Fundamentals of Model-Driven Engineering: Mischform einer Frontalveranstaltung (kurze Vorlesungseinheiten) und Übung. Übungsaufgaben werden während der Veranstaltung in Gruppen bearbeitet und diskutiert. Jede Übung besteht aus einem theoretischen und einem praktischen Teil. Der praktische Teil baut auf einem in Java programmierten Framework auf, das für die Vorlesung konzipiert und implementiert wurde.</p> <p>Die letzten 4-5 Termine der Veranstaltung werden für eine themenspezifische Vertiefung der Grundinhalte verwendet. Es werden Gruppen von 3-4 Studierenden gebildet, weiterführende Themen besprochen, gewählt, und als "Mini-Praktikum" parallel bearbeitet. Ergebnisse werden bei dem letzten Termin allen anderen Gruppen präsentiert. In dieser letzten Phase der Veranstaltung sollen Studierende vor allem die selbständige Einarbeitung und Erarbeitung von neuem weiterführendem Wissen üben.</p>
<b>Prüfungsleistung (Dauer)</b>
<p>Mündliche Prüfung (ca. 40 Minuten)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: schriftliche Übungsaufgaben  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.



<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Jun.-Prof.Dr. Anthony Anjorin
<b>Lernmaterialien, Literaturangaben</b>
Fundamentals of Model-Driven Engineering: Ehrig, H., Ehrig, K., Prange, U., & Taentzer, G. (2006). Fundamentals of Algebraic Graph Transformation. (W. Brauer, G. Rozenberg, & A. Salomaa, Eds.). Springer. Awodey, S. (2006). Category Theory. Ebsco Publishing.
<b>Sonstige Hinweise</b>
keine

### 3.22 Wahlpflichtmodul: Future Internet

Modulname	Future Internet / Future Internet
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Future Internet : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Future Internet: Vorlesung ( 45h / 105h / EN / SS / 20 ) Future Internet: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Future Internet: Grundkenntnisse in Rechnernetze (z.B. durch die Bachelor-Vorlesung Rechnernetze).	
<b>Inhalte</b>	
Future Internet: Die Veranstaltung diskutiert aktuelle, forschungsnahe Entwicklung des Internets und der Vernetzung von Rechenzentren. Sie wird dynamisch an entsprechende Themen angepasst und basiert insbesondere auf wissenschaftlichen Veröffentlichungen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer werden durch die Vorlesung an den aktuellen Stand der Internet-Forschung herangeführt. Sie kennen die Schwachpunkte der aktuellen Architektur, können diese geeignet kritisieren und können diese mit aktuellen Vorschlägen kontrastieren sowie Vor- und Nachteile der jeweiligen Lösungen bewerten. Sie können für unterschiedliche Nutzungssituationen die Anwendbarkeit einer bestimmten Lösung einschätzen und voraussagen. Methodisch sind sie in der Lage, Netz-Experimente zu entwerfen und durchzuführen. Teilnehmer können neue Vorschläge für Architekturen und Protokolle kreieren, diese mit anderen Ansätzen vergleichen und bewerten, und sich für eine geeignete Lösung entscheiden. Da die Vorlesung auf aktuellen Veröffentlichungen beruht, sind Teilnehmer in der Lage, sich selbständig in neues, nicht didaktisch aufbereitetes Material einzuarbeiten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	

Future Internet: Vorlesung mit Folien und Tafelanschrieb; begleitende Übungen. Teilnehmer werden zu eigener Literaturstudie aktueller Veröffentlichungen angehalten. In der Übung werden Architekturexperimente, bspw. mit OpenFlow, durchgeführt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Future Internet: Foliensatz, insbesondere auch aktuelle Veröffentlichungen. Kein umfassendes Lehrbuch verfügbar, Teile abgedeckt durch Stallings, Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud.
<b>Sonstige Hinweise</b>
keine

### 3.23 Wahlpflichtmodul: Hardware/Software Codesign

Modulname	Hardware/Software Codesign / Hardware/Software Codesign
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Hardware/Software Codesign : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Hardware/Software Codesign: Vorlesung ( 45h / 105h / EN / SS / 40 ) Hardware/Software Codesign: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Hardware/Software Codesign: Digitaltechnik, Rechnerarchitektur, Programmierkenntnisse	
<b>Inhalte</b>	
Hardware/Software Codesign: Hardware/Software Codesign bezeichnet den integrierten und automatisierten Entwurf von Hard- und Software in Computersystemen. Die Lehrveranstaltung Hardware/Software Codesign lehrt Konzepte und Methoden, welche in computerunterstützten Entwurfswerkzeugen zur Entwurfsraumexploration, Entwurfsoptimierung und Compilation für spezialisierte Computersysteme verwendet werden.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer dieser Lehrveranstaltung können die Ziele und Herausforderungen beim Entwurf von spezialisierten Computersystemen benennen. Sie können die passenden Modellierungsansätze für ein gegebenes HW/SW System und entsprechenden funktionalen und nichtfunktionalen Anforderungen auswählen und eine Spezifikation im gewählten Formalismus erstellen. Sie können Zielarchitekturen für die Implementierung von HW/SW Systemen charakterisieren und die Eignung einer spezifischen Zielarchitektur für eine gegebene Anwendung bewerten. Sie können den Aufbau eines Compilers beschreiben, Grundblöcke und Kontrollflussgraphen verstehen und anwenden, sowie Optimierungs- und Codegenerierungsmethoden diskutieren. Sie können demonstrieren, wie Programme mittels High-level Synthesemethoden in Hardware übersetzt werden. Sie verstehen die Methode der ganzzahlig linearen Programmierung und können sie auf Probleme aus den Bereichen Synthese, Ablaufplanung und Software Performanceschätzung anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	

<b>Methodische Umsetzung</b>
Hardware/Software Codesign: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Selbststudium und Diskussion wissenschaftlichen Publikationen</li> <li>• Praktische Programmierprojekte zur Umsetzung und Anwendung von Methoden aus dem Bereich Hardware/Software Codesign</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Christian Plessl
<b>Lernmaterialien, Literaturangaben</b>
Hardware/Software Codesign: Folien, Standardlehrbücher, Übungsblätter, Wissenschaftliche Artikel
<b>Sonstige Hinweise</b>
keine

### 3.24 Wahlpflichtmodul: High-Performance Computing

Modulname	High-Performance Computing / High-Performance Computing
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• High-Performance Computing : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
High-Performance Computing: Vorlesung ( 30h / 105h / EN / WS / 40 ) High-Performance Computing: Übung ( 45h / 0h / EN / WS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
High-Performance Computing: <ul style="list-style-type: none"> <li>• Programmierkenntnisse in C/C++</li> <li>• Rechnerarchitektur</li> </ul>	
<b>Inhalte</b>	
High-Performance Computing: Die Veranstaltung vermittelt Grundlagen des Hochleistungsrechnen (High-Performance Computing) mit einem Schwerpunkt auf der Programmierung von parallelen Rechensystemen und neuartiger Hardwarebeschleuniger.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• Modelle und Programmiermuster für HPC zu benennen und die passenden Muster für eine gegebenen Anwendung zu identifizieren,</li> <li>• die Grundkonstrukte der gängigen HPC Bibliotheken, insbesondere MPI, OpenMP und OpenCL, anzugeben und anzuwenden,</li> <li>• die Performance von Anwendungen durch Verwendung von Profilingwerkzeugen zu analysieren und systematisch passende Optimierungsstrategien abzuleiten,</li> <li>• die gelernten Konzepte und Verfahren auf existierende Anwendungen anzuwenden und diese zu parallelisieren und optimieren.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> </ul>	
<b>Methodische Umsetzung</b>	
High-Performance Computing:	

<ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Praktische Programmierprojekte auf Parallelrechnersystemen in Kleingruppen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Christian Plessl
<b>Lernmaterialien, Literaturangaben</b>
High-Performance Computing: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Programmierprojekte</li> <li>• Lehrbuch: Pacheco: An Introduction to Parallel Programming. Morgan Kaufmann, 2011.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.25 Wahlpflichtmodul: Intelligence in Embedded Systems

Modulname	Intelligence in Embedded Systems / Intelligence in Embedded Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Intelligenz in eingebetteten Systemen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Intelligenz in eingebetteten Systemen: Vorlesung ( 45h / 105h / EN / SS / 25 ) Intelligenz in eingebetteten Systemen: Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Intelligenz in eingebetteten Systemen:	
<b>Inhalte</b>	
<p>Intelligenz in eingebetteten Systemen: Intelligente eingebettete Systeme sind technische Systeme, die mittels unterschiedlicher Sensoren und Aktoren ihre Umwelt wahrnehmen sowie (teil-)autonom mit ihr interagieren. Häufig werden Methoden und Algorithmen der Künstlichen Intelligenz zur Steuerung des Verhaltens eingesetzt. Diese ermöglichen es den Systemen beispielsweise, ihr Verhalten zielgerichtet zu planen sowie durch Adaption und Lernen selbständig zu optimieren. Intelligente eingebettete Systeme gewinnen kontinuierlich an Bedeutung, nicht nur auf wissenschaftlicher Ebene innerhalb der Informatik, sondern auch im sozialen und gesellschaftlichen Kontext: Autonome oder teilautonome Systeme wie Serviceroboter, selbstfahrende PKWs oder medizinische Hilfs- und Diagnosesysteme werden unser privates und berufliches Leben in absehbarer Zukunft tiefgreifend verändern.</p> <p>Diese Vorlesung greift wichtige Aspekte des Entwurfs intelligenter eingebetteter Systeme auf und vermittelt entsprechende theoretische und methodische Grundlagen. Ausgehend von typischen Architekturen solcher Systeme erstrecken sich die Inhalte über Themen der intelligenten Sensorverarbeitung und Modellierung der Umwelt bis hin zur intelligenten Handlungssteuerung und Selbstadaption.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende benennen und erklären Methoden und Algorithmen zur intelligenten Sensorverarbeitung und Handlungssteuerung (z. B. Bildverarbeitung, Sensorfusion, Kartendarstellung, Navigation, Planung und maschinellem Lernen). Sie verstehen und lösen Probleme bei der Umsetzung in eingebetteten Systemen, die über eingeschränkte Ressourcen verfügen. Ferner sind sie in der Lage sich in neue Verfahren einzuarbeiten, sie zu beurteilen und sie einzusetzen, insbesondere im Kontext eingebetteter Systeme.	
<b>Nichtkognitive Kompetenzen</b>	



<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Intelligenz in eingebetteten Systemen: <ul style="list-style-type: none"> <li>• Vorlesung mit Folien</li> <li>• Interaktive Übungen, in denen die Studenten das Verständnis des Stoffes vertiefen und das Gelernte anwenden</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Bernd Kleinjohann
<b>Lernmaterialien, Literaturangaben</b>
Intelligenz in eingebetteten Systemen: Folien, Publikationen, Bücher: <ul style="list-style-type: none"> <li>• St. Russel, P. Norvig: Artificial Intelligence: A Modern Approach</li> <li>• R. Arkin: Behavior-Based Robotics</li> </ul> Weitere Literatur (Bücher, Publikationen) werden in der Vorlesung bekanntgegeben.
<b>Sonstige Hinweise</b>

### 3.26 Wahlpflichtmodul: Interactive Data Visualization

Modulname	Interactive Data Visualization / Interactive Data Visualization
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Interaktive Datenvisualisierung : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Interaktive Datenvisualisierung: Vorlesung ( 30h / 105h / EN / SS / 30 ) Interaktive Datenvisualisierung: Übung ( 15h / 0h / EN / SS / 30 ) Interaktive Datenvisualisierung: Visualisierungsprojekt ( 30h / 0 h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Interaktive Datenvisualisierung: Sicherer Umgang mit zumindest einem Grafik API.	
<b>Inhalte</b>	
<p>Interaktive Datenvisualisierung: Die Datenvisualisierung stellt Methoden und Techniken bereit, um die den Datensätzen zugrunde liegenden korrelierenden Strukturen und Beziehungen zu erfassen und zu präsentieren. Datensätze können dabei aus unterschiedlichsten Anwendungen kommen. Das Hauptziel der Präsentation ist die Kommunikation der im Datensatz enthaltenen Information in ausdrucksvoller und wirksamer Weise für den menschlichen Betrachter. In diesem Kontext ist die Visualisierung (=das Ergebnis des Visualisierungsprozesses) ein computergeneriertes Bild oder eine Folge von Bildern, wobei der Input in diesen Prozess Daten sind und das Ziel dieses Prozesses die menschliche Wahrnehmung ist. Der Visualisierungsprozess kann also auch als Abbildung von Daten in ausdrucksstarke und wirksame (visuelle) Repräsentationen für den Menschen gesehen werden.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Der Visualisierungsprozess kann von Studierenden anhand des Vier-Komponentenmodells (Reality, Data, Picture(s), User) erklärt werden. Studierende besitzen die Fähigkeit, selektive Algorithmen zur Datenanalyse und Datenvorverarbeitung (z.B. Transferfunktionen, Filterung, statistische Analysen, Sampling, Skalierung) in Rechenschritten nachzuvollziehen und mit einem modernen API umzusetzen. APIs und Tools zur Visualisierung von unterschiedlichen Datensätzen und Visualisierungszielen können bewertet werden. Für unterschiedliche Datenmodelle können mehrere mögliche Visualisierungstechniken und Interaktionstechniken benannt und erklärt werden. Studierende besitzen die Fähigkeit, ihre eigenen sowie fremde Visualisierungen auf ihre Ausdrucksfähigkeit, Wirksamkeit, Angemessenheit und Skalierbarkeit zu bewerten. Bewertungsmethoden (Evaluationsmethoden) für den Einsatz in der Entwicklung von Visualisierungssoftware können benannt und erklärt werden. Über den Design Prozess in der Visual-</p>	

<p>isierung (Abbildung von Daten-Parametern auf visuelle Variablen) kann qualifiziert diskutiert werden. Für den Visualisierungsprozess relevante Nutzer-, Daten-, und Hardwareeigenschaften sind bekannt und können in den Design Prozess eingebracht werden. Studierende demonstrieren in einem Visualisierungsprojekt ihre Fähigkeit, sich in</p> <ul style="list-style-type: none"> <li>• Anwendungskontexte (z.B. Medizin, Astronomie, Computer-Netzwerk-Analyse, Textkorpora-Analyse) schnell einzuarbeiten,</li> <li>• alleine oder im Team Lösungswege zu recherchieren, zu verstehen, und einen oder mehrere Lösungswege zu implementieren,</li> <li>• Lösungswege und resultierende Visualisierungen technisch verständlich und motivierend zu präsentieren.</li> </ul>
<p><b>Nichtkognitive Kompetenzen</b></p>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Medienkompetenz</li> </ul>
<p><b>Methodische Umsetzung</b></p>
<p>Interaktive Datenvisualisierung: Die Vorlesung nutzt Beamer und Tafel. Die Studierenden bearbeiten kurze In-Class Aufgaben und diskutieren dann mit der Dozentin über unterschiedliche Lösungen bzw. Probleme bei den Lösungen. Hausaufgaben werden nach kurzer Vorbereitung in der Zentralübung von den Studierenden alleine gelöst, ihre Lösungen werden in der Zentralübung vorgestellt und diskutiert. Im letzten Drittel der Veranstaltung wird an einem Gruppenprojekt gearbeitet. Die Studierenden präsentieren am Ende der Veranstaltung ihre Lösungen und bewerten die Projekte ihrer Kommilitonen.</p>
<p><b>Prüfungsleistung (Dauer)</b></p>
<p>Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<p><b>Moduleilprüfungen</b></p>
<p>keine</p>
<p><b>Studienleistung / qualifizierte Teilnahme</b></p>
<p>Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<p><b>Voraussetzungen für die Teilnahme an der Prüfung</b></p>
<p>Bestehen der Studienleistung.</p>
<p><b>Voraussetzungen für die Vergabe von Credits</b></p>
<p>Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</p>
<p><b>Gewichtung für die Gesamtnote</b></p>
<p>Das Modul wird mit 6 Credits gewichtet.</p>

<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gitta Domik-Kienegger
<b>Lernmaterialien, Literaturangaben</b>
Interaktive Datenvisualisierung: Foliensatz der Vorlesung; Textbuch "Interactive Data Visualization", M. Ward, G. Grinstein, D. Keim, A K Peters.
<b>Sonstige Hinweise</b>
keine

### 3.27 Wahlpflichtmodul: Kontextuelle Informatik

Modulname	Kontextuelle Informatik / Contextual Informatics
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Kontextuelle Informatik : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Kontextuelle Informatik: Vorlesung ( 30h / 105h / DE / WS / 40 ) Kontextuelle Informatik: Übung ( 45h / 0h / DE / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Kontextuelle Informatik: Kenntnisse aus dem Bachelorstudiengang Informatik oder eines vergleichbaren Studiengangs	
<b>Inhalte</b>	
<p>Kontextuelle Informatik: Informatiker entwickeln auf Zeichen basierende Produkte (Programme, Spezifikationen, Dokumentationen etc.), die einen spezifischen Gegenstandsbereich modellieren. Bei der Entwicklung solcher Produkte stellen sich vielfältige Fragen: Wie können die zu verarbeitenden Daten sowie die umzusetzenden Prozesse angemessen modelliert werden? Welche Konsequenzen ergeben sich aus der Möglichkeit, Systeme interaktiv zu gestalten? Welche Rolle wird beim Einsatz der Software den Benutzern, welche der Software zuteil? Welche Rahmenbedingungen des Einsatzkontexts sind dabei zu beachten?</p> <p>Die Veranstaltung »Kontextuelle Informatik« erörtert die im weiteren Verlauf relevanten Grundbegriffe der Informatik mit besonderem Augenmerk auf die Unterscheidung zwischen technischen Konzepten und der Nutzungssphäre. Vor diesem Hintergrund werden Theorien interaktiver Systeme betrachtet, um insbesondere die Rolle digitaler Medien für geistige Prozesse untersuchen. Bei der Entwicklung von Informatiksystemen müssen die relevanten Daten und Prozesse bis zu einem gewissen Grad antizipiert und als formales System beschrieben werden. Dies wirft Fragen auf, unter welchen Bedingungen eine solche formale Beschreibung adäquat erfolgen kann und welche Konsequenzen sich daraus in Bezug auf die Zuverlässigkeit und den verantwortbaren Einsatz von Informatiksystemen ergeben.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden lernen, theoriegeleitet die Bedeutung von interaktiven Systemen zu untersuchen. Sie verstehen, technische und nicht-technische Problemstellungen zu differenzieren und adäquat aufeinander zu beziehen. Des Weiteren werden sie in die Lage versetzt, aktuelle technologische Entwicklungen zu	

bewerten und zu vergleichen sowie Innovationspotenziale im Bereich digitaler Medien abschätzen zu können.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Medienkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>
<b>Methodische Umsetzung</b>
Kontextuelle Informatik: Vorlesungsinhalte werden im Vortragsstil vermittelt, wobei die Studierenden durch Fragen und kurze Diskussionen interagieren. In den Übungen werden die Inhalte auf der Basis der Lektüre wissenschaftlicher Literatur sowie mit Kurzreferaten durch die Studierenden vertieft.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Harald Selke
<b>Lernmaterialien, Literaturangaben</b>
Kontextuelle Informatik: Vorlesungsfolien Wardrip-Fruin, N.; Montfort, N. (eds.): The New Media Reader. Cambridge, Ma.: MIT Press, 2003. Quinn, M. J.: Ethics for the Information Age. Boston, Ma.: Pearson, 7th edition, 2016. Begleitende wissenschaftliche Literatur wird in der Vorlesung vorgestellt.

<b>Sonstige Hinweise</b>
keine

### 3.28 Wahlpflichtmodul: Language-Based Security

Modulname	Language-Based Security / Language-Based Security
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Sprachbasierte Sicherheit : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Sprachbasierte Sicherheit: Vorlesung ( 45h / 105h / EN / WS / 25 ) Sprachbasierte Sicherheit: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Sprachbasierte Sicherheit: Grundlegende Kenntnisse aus den Bereichen Compilerbau, Programmiersprachen und IT Sicherheit. (Aus den jeweiligen Bachelor Vorlesungen “Programming Languages and Compilers” und “IT Sicherheit”.)	
<b>Inhalte</b>	
Sprachbasierte Sicherheit: Ziel des Kurses ist es, Grundlagen aus der sprach-basierten Sicherheit aus praktischer und theoretischer Sicht zu vermitteln.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studenten erwerben fundierte Kenntnisse zu aktuellen Angriffen und Verteidigungstechniken. Behandelte Techniken werden sowohl theoretisch als auch praktisch behandelt, sodass Studenten neben Faktenwissen zu den jeweiligen Techniken auch jene Methodenkompetenzen erwerben, die es ihnen erlaubt Sicherheitsfragestellungen aus Programmiersprachen-Sicht kompetent zu beantworten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Sprachbasierte Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb.</li> <li>• Begleitende Implementierung als Übungsteil.</li> </ul>	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	



Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof.Dr. Stefan Brunthaler
<b>Lernmaterialien, Literaturangaben</b>
Sprachbasierte Sicherheit: keine
<b>Sonstige Hinweise</b>
keine

### 3.29 Wahlpflichtmodul: Linear and Integer Optimization

Modulname	Linear and Integer Optimization / Linear and Integer Optimization
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Linear and Integer Optimization : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Linear and Integer Optimization: Vorlesung ( 45h / 105h / EN / WS / 30 ) Linear and Integer Optimization: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Linear and Integer Optimization: Algorithmen und Datenstrukturen Lineare Algebra	
<b>Inhalte</b>	
<p>Linear and Integer Optimization: Optimierungsprobleme sind heute allgegenwärtiger Bestandteil des täglichen Lebens in Industrie, Handwerk und Handel. Dabei geht es zumeist um die optimale Nutzung teurer Ressourcen, z.B. bei Transport-, Scheduling-, Planungs- oder Designproblemen. Viele dieser Optimierungsprobleme sind NP-hart und daher nicht effizient lösbar. Überraschenderweise lassen sich viele dieser Optimierungsprobleme anschaulich als lineare Programme (LP) oder ganzzahlig-lineare Programme (IP) modellieren und die Algorithmen und Techniken zur Lösung von LPs und ILPs zeigen hervorragende Ergebnisse bei der Lösung dieser Modelle.</p> <p>Die Vorlesung gibt eine grundlegende Übersicht über lineare und ganzzahlig-lineare Optimierung. Dabei behandeln wir die Modellierung als LP/IP, Algorithmen zur exakten Lösung von LPs / IPs, betrachten effizient lösbare Spezialfälle von IPs und Heuristiken zur Steigerung der Effizienz der Löser im allgemeinen Fall.</p> <p>Stichworte: Simplexalgorithmus, Innere-Punkte-Verfahren, Schnittebenenverfahren, Branch &amp; Bound-Algorithmus, Branch &amp; Cut-Verfahren</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende lernen grundlegende Modellierungstechniken für Optimierungsprobleme. Sie können erkennen, welche Modellierungen für ein Anwendungsproblem geeignet sind und ob sich die erstellten Modelle effizient lösen lassen. Für lineare und ganzzahlige Programme kennen die Studierenden grundlegende Algorithmen zur Lösung sowie unterschiedliche Techniken zur Effizienzsteigerung. Desweiteren können Studierende kommerzielle und nichtkommerzielle Software zur Lösung von LPs und IPs einsetzen.	
<b>Nichtkognitive Kompetenzen</b>	

<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Linear and Integer Optimization: Vorlesung mit Übung
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Rainer Feldmann
<b>Lernmaterialien, Literaturangaben</b>
Linear and Integer Optimization: Folien, Skript Literatur: M. Conforti, G. Cornuejols, G. Zambelli. Integer Programming. Springer 2014. A. Schrijver. Combinatorial Optimization: Polyhedra and Efficiency, Vol. A,B,C. Springer 2003. V. Chvatal. Linear Programming. Freeman 1983. L.A. Nemhauser, G.L. Wolsey. Integer and Combinatorial Optimization. Wiley 1999. A. Schrijver. Theory of Linear and Integer Optimization. Wiley, 1999.
<b>Sonstige Hinweise</b>
keine

### 3.30 Wahlpflichtmodul: Logic and Automated Reasoning

Modulname	Logic and Automated Reasoning / Logic and Automated Reasoning
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Logik und automatisches Beweisen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Logik und automatisches Beweisen: Vorlesung ( 45h / 105h / EN / WS / 30 ) Logik und automatisches Beweisen: Übung ( 30h / 0h / EN / WS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Logik und automatisches Beweisen: Grundlagen symbolischer Modellierung, Grundlagen zur Komplexitätstheorie	
<b>Inhalte</b>	
Logik und automatisches Beweisen: Die Veranstaltung "Logik und automatisches Beweisen" soll verschiedene Aspekte der Argumentation und Beweisführung verständlich machen. Basis ist eine symbolische Repräsentation von Wissen über Problemdomänen. Für die Wissensrepräsentation werden logikbasierte Sprachen (klassisch und nicht-klassisch) vorgestellt und ihre Ausdrucksfähigkeit untersucht. Für diese Sprachen werden jeweils passende Deduktionsansätze besprochen und analysiert. Insbesondere werden Zusammenhänge zwischen aussagenlogischen und prädikatenlogischen Ansätzen beleuchtet, sowie spezielle Fragestellungen wie Korrektheit, (Un-) Vollständigkeit und (Nicht-) Monotonie.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage <ul style="list-style-type: none"> <li>• logikbasierte Sprachen zur Wissensrepräsentation zu benennen, ihre Eigenschaften zu beschreiben und zu bewerten,</li> <li>• Wissen und Aufgabenstellungen aus einfachen Domänen mit Hilfe eines logikbasierten Formalismus symbolisch zu repräsentieren,</li> <li>• verschiedene Kalküle und zugehörige Inferenzverfahren zu beschreiben,</li> <li>• geeignete Inferenzverfahren auszuwählen und anzuwenden,</li> <li>• die Komplexität der Aufgabenstellung und Laufzeit von Inferenzverfahren anhand einfacher Skalen zu beurteilen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Logik und automatisches Beweisen: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Übungen</li> <li>• Diskussion</li> <li>• Hausaufgaben</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Theodor Lettmann
<b>Lernmaterialien, Literaturangaben</b>
Logik und automatisches Beweisen: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Lehrbücher: 1.) J. Harrison: Handbook of Practical Logic and Automated Reasoning, Cambridge University Press, 2009, 2.) M. Fitting: First-Order Logic and Automated Theorem Proving, 2nd ed., Springer 1995, 3.) j.H. Gallier: Logic for Computer Science: Foundations of Automatic Theorem Proving, 2nd ed., Dover, 2015, 4.) L. Wos, G. W. Pieper. A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning, World Scientific, 1999.</li> </ul>

- Übungsaufgaben
- Liste von klassischen und aktuellen Papieren, z.B. D. W. Loveland. Mechanical theorem-proving by model elimination, Journal of the ACM,15, pp. 236–251, 1968

**Sonstige Hinweise**

keine

### 3.31 Wahlpflichtmodul: Logic Programming for Artificial Intelligence

Modulname	Logic Programming for Artificial Intelligence / Logic Programming for Artificial Intelligence
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Logic Programming for Artificial Intelligence : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Logic Programming for Artificial Intelligence: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Logic Programming for Artificial Intelligence: Übung ( 30h / 0h / EN / WS oder SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Logic Programming for Artificial Intelligence: Studenten sollten Vorkenntnisse in der Programmierung haben, wie sie in den Lehrveranstaltungen "Programmierung und GPS" angeboten werden, sowie Kenntnisse in Datenbank-Anfragesprachen wie sie in der Lehrveranstaltung "Datenbanksysteme" angeboten werden.	
<b>Inhalte</b>	
Logic Programming for Artificial Intelligence: Diese Lehrveranstaltung betrachtet verschiedene Konzepte und Techniken der Informatik, der Künstlichen Intelligenz und der Computerlinguistik aus einer anderen Perspektive, aus der Perspektive der Logikprogrammierung. Logikprogrammierung im Allgemeinen und die Programmiersprache Prolog im Besonderen erlauben es, viele Konzepte deklarativ in Logik zu beschreiben und gleichzeitig durch einen Interpreter zu testen und auszuführen. Dies eignet sich besonders für die Lösung von Puzzle- und Quiz-Aufgaben, aber auch für selbst definierte und Domänen-spezifische Sprachen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Studenten lernen</p> <p>Faktenwissen über</p> <ul style="list-style-type: none"> <li>• die Transformation von Wissen, das als Fakten und Regeln gegeben ist, in ausführbare Programme</li> <li>• die Programmierung in Logik und in selbst entworfenen Sprachen</li> </ul> <p>Methodisches Wissen, unter anderem die Fähigkeit:</p> <ul style="list-style-type: none"> <li>• Domänen-spezifische Sprachen zu definieren</li> <li>• Interpreter für Domänen-spezifische Sprachen zu implementieren</li> <li>• kleine Frage-Antwort-Systeme zu implementieren</li> </ul>	

<ul style="list-style-type: none"> <li>• Software zu entwickeln für Theorembeweiser, Constraint-Solver und zur Lösung von Puzzles</li> </ul> <p>Transfer-Wissen, u.a. die Fähigkeit:</p> <ul style="list-style-type: none"> <li>• die Methoden auf neue Probleme, Kalküle und Wissensrepräsentationsformate zu übertragen</li> <li>• das Wissen über Parsing und Semantik-Konstruktion auf Domänen-spezifische Sprachen zu übertragen</li> </ul> <p>Normatives Wissen, unter anderem die Fähigkeit folgendes zu bewerten:</p> <ul style="list-style-type: none"> <li>• die Eignung und die Grenzen verschiedener Daten- und Wissensrepräsentationsformate für verschiedene Aufgaben</li> <li>• die Eignung verschiedener Programmier-Paradigmen für verschiedene Projekte</li> <li>• den Aufwand und die Machbarkeit von Projekten mit dem Ziel natürliche Sprache zu verstehen</li> <li>• den Aufwand und die Machbarkeit von Projekten mit dem Ziel natürliche Sprache zu übersetzen</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Logic Programming for Artificial Intelligence: Theoretische Konzepte werden in der Vorlesung erläutert und in den Tutorien in kleinen Gruppen vertieft. Tutorien werden als praktische Übungen am Computer durchgeführt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–



<b>Modulbeauftragte/r</b>
Prof. Dr. Stefan Böttcher
<b>Lernmaterialien, Literaturangaben</b>
Logic Programming for Artificial Intelligence: Ivan Bratko: Prolog Programming for Artificial Intelligence. Pearson Education, Newest Edition. Hinweise auf weiteres Material werden in der Lehrveranstaltung bekannt gegeben.
<b>Sonstige Hinweise</b>
keine

### 3.32 Wahlpflichtmodul: Machine Learning I

Modulname	Machine Learning I / Machine Learning I
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Maschinelles Lernen I : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Maschinelles Lernen I: Vorlesung ( 45h / 105h / EN / WS oder SS / 60 ) Maschinelles Lernen I: Übung ( 30h / 0h / EN / WS oder SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Maschinelles Lernen I: Grundlagen in Mathematik (lineare Algebra, Statistik), Programmierung und Algorithmen.	
<b>Inhalte</b>	
Maschinelles Lernen I: Aufgrund der stetig wachsenden Menge an Daten, die in unserer Informationsgesellschaft systematisch produziert wird, hat das Maschinelle Lernen in den letzten Jahren mehr und mehr an Bedeutung gewonnen, nicht nur als wissenschaftliche Disziplin sondern auch als Schlüsseltechnologie für moderne Software und intelligente Systeme. Diese Vorlesung gibt eine Einführung in das Maschinelle Lernen, wobei der Fokus auf dem überwachten Lernen für Klassifikation und Regression liegt. Theoretische Grundlagen der Generalisierung werden ebenso behandelt wie praktische Aspekte und konkrete Lernalgorithmen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden verstehen die statistischen Grundlagen der Generalisierung, d.h. der Induktion von Modellen aus Daten, sowie praktischen Ansätzen zur Modellvalidierung. Sie können grundlegende Methoden und Algorithmen des überwachten Lernens auf Klassifikations- und Regressionsprobleme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	

Maschinelles Lernen I: Theoretische Grundlagen und Konzepte des Maschinellen Lernens werden im Rahmen einer Vorlesung eingeführt und anschließend in praktischen Übungen in Kleingruppen sowie in Heimübungen vertieft ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eyke Hüllermeier
<b>Lernmaterialien, Literaturangaben</b>
Maschinelles Lernen I: Skript und eine Liste mit Buchempfehlungen: <ul style="list-style-type: none"> <li>• Y.S. Abu-Mostafa, M. Magdon-Ismael, H.T. Lin. Learning from Data, AMLBook, 2012.</li> <li>• P. Flach. Machine Learning, Cambridge Univ. Press, 2012.</li> <li>• E. Alpaydin. Machine Learning, Oldenbourg, 2008.</li> <li>• C.M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.33 Wahlpflichtmodul: Machine Learning II

Modulname	Machine Learning II / Machine Learning II
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Maschinelles Lernen II : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Maschinelles Lernen II: Vorlesung ( 45h / 105h / EN / WS oder SS / 20 ) Maschinelles Lernen II: Übung ( 30h / 0h / EN / WS oder SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Maschinelles Lernen II: Grundlegende Kenntnisse in Maschinellern Lernen (z.B. vermittelt durch die Machine Learning I Vorlesung).	
<b>Inhalte</b>	
Maschinelles Lernen II: Aufbauend auf einer grundlegenden Einführung in das maschinelle Lernen, wie beispielsweise vermittelt durch die Veranstaltung Machine Learning I, werden in dieser Vorlesung fortgeschrittene Themen in diesem Gebiet behandelt (reinforcement learning, online learning and bandit algorithms, multi-task learning, multi-target and structured output prediction, preference learning, learning from weak supervision, and uncertainty in machine learning). Obwohl die Vorlesung im Wesentlichen methodisch und algorithmisch ausgerichtet ist, werden auch theoretische und anwendungsorientierte Aspekte behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden haben einen Überblick über Methoden zur Klassifikation mit mehr als zwei Klassen, dem Lernen nichtlinearer Modelle, sowie Erweiterungen des einfachen Szenarios des überwachten Lernens. Sie verstehen algorithmische Konzepte entsprechender Lernverfahren und können diese Verfahren auf praktische Probleme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	

Maschinelles Lernen II: Theoretische Grundlagen und Konzepte des Maschinellen Lernens werden im Rahmen einer Vorlesung eingeführt und anschließend in praktischen Übungen in Kleingruppen sowie in Heimübungen vertieft ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eyke Hüllermeier
<b>Lernmaterialien, Literaturangaben</b>
Maschinelles Lernen II: Skript und eine Liste mit Buchempfehlungen: <ul style="list-style-type: none"> <li>• Y.S. Abu-Mostafa, M. Magdon-Ismail, H.T. Lin. Learning from Data, AMLBook, 2012.</li> <li>• P. Flach. Machine Learning, Cambridge Univ. Press, 2012.</li> <li>• E. Alpaydin. Machine Learning, Oldenbourg, 2008.</li> <li>• C.M. Bishop. Pattern Recognition and Machine Learning, Springer, 2006.</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.34 Pflichtmodul: Master-Abschlussarbeit

Modulname	Master-Abschlussarbeit / Master Thesis
Workload	900 h
Leistungspunkte	30 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Masterarbeit Arbeitsplan : 4</li> <li>• Masterarbeit : 4</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Masterarbeit Arbeitsplan: Kontaktzeiten, Präsentation Arbeitsplan ( 30h / 120 h / EN / WS oder SS / 0 ) Masterarbeit: Kontaktzeiten, Ergebnispräsentation ( 30h / 720 h / EN / WS oder SS / 1 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Modulprüfungen im Hauptfach im Umfang von 48 LP müssen erfolgreich abgelegt worden sein.	
<b>Empfohlene Kenntnisse</b>	
Masterarbeit Arbeitsplan: Je nach gewähltem Thema. Masterarbeit: Je nach gewähltem Thema.	
<b>Inhalte</b>	
Masterarbeit Arbeitsplan: Nach Themenabsprache mit dem Betreuer erfolgt eine erste grobe Einarbeitung. Auf dieser Grundlage und einer ersten Literaturrecherche ist durch den Studierenden ein Arbeitsplan vorzulegen, der die zu erzielenden Ergebnisse samt Meilensteine für die Arbeit dokumentiert.  Masterarbeit: In der Masterarbeit zeigt der/die Kandidat/in seine/ihre Fähigkeit zum selbständigen wissenschaftlichen Arbeiten an einem angemessen anspruchsvollen Thema, das auch Gelegenheit zur Entfaltung eigener Ideen gibt. Auf der Grundlage des State-of-the-art sollen die Methoden der Informatik systematisch angewendet werden. Die Ergebnisse der Arbeit werden in einem fakultätsöffentlichen Vortrag vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Mit Abschluss der Masterarbeit haben die Studierenden gezeigt, dass sie <ul style="list-style-type: none"> <li>• ein Problem innerhalb einer bestimmten Frist ein Problem nach wissenschaftlichen Methoden bearbeiten können,</li> <li>• die im Zuge des Studiums erworbenen fachlich-methodischen fachübergreifenden Kompetenzen zur Lösung komplexer Aufgabenstellungen anwenden können. Die Schwerpunkte variieren ja nach Aufgabenstellung.</li> </ul>	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Masterarbeit Arbeitsplan: Direkte Absprache mit Betreuer.</p> <p>Masterarbeit: Selbständiges Arbeiten unterstützt durch individuelle Betreuung</p>
<b>Prüfungsleistung (Dauer)</b>
<p>Abschlussarbeit</p> <p>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 50 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
<p>Masterarbeit Arbeitsplan: Je nach gewähltem Thema.</p> <p>Masterarbeit: Je nach gewähltem Thema.</p>
<b>Sonstige Hinweise</b>
Die Masterarbeit ist eine Prüfungsarbeit, die die wissenschaftliche Ausbildung abschließt und zeigen soll, dass die Kandidatin oder der Kandidat die Fähigkeit besitzt, innerhalb einer bestimmten Frist ein

Problem der Informatik nach wissenschaftlichen Methoden zu bearbeiten. Die Aufgabenstellung soll so gestaltet werden, dass sie einem Arbeitsaufwand von fünf Monaten Vollzeitarbeit entspricht. Die Arbeit muss fünf Monate nach der Ausgabe abgegeben werden. Die Arbeit soll einen Umfang von in der Regel nicht mehr als 120 DIN A4-Seiten haben.



### 3.35 Wahlpflichtmodul: Mobile Communication

Modulname	Mobile Communication / Mobile Communication
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Mobilkommunikation : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Mobilkommunikation: Vorlesung ( 45h / 105h / EN / WS / 20 ) Mobilkommunikation: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Mobilkommunikation: Grundkenntnisse in Rechnernetze (z.B. durch die Bachelor-Vorlesung Rechnernetze).	
<b>Inhalte</b>	
Mobilkommunikation: Die Veranstaltung behandelt grundlegende Techniken für die Mobilkommunikation (z.B. drahtlose Kanalmodelle) und Techniken (z.B. Spreizbandkommunikation), wesentliche Protokollmechanismen (z.B. Medienzugriff), Systeme der Mobilkommunikation sowie MobileIP. Neben technologischen und konzeptionellen Aspekten werden auch Verfahren und Methoden zur Leistungsbewertung besprochen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Teilnehmer kennen die Herausforderungen und Probleme beim Entwurf und Betrieb von Mobilkommunikationssystemen. Sie können zwischen physikalischen und entwurfsbedingten Problemstellungen differenzieren und geeignete Protokollmuster auswählen bzw. neue Protokolle konstruieren. Sie sind in der Lage, Mechanismen unterschiedlicher Architekturebenen auszuwählen, in eine sinnvolle Gesamtarchitektur zu integrieren und diese Auswahl zu begründen. Sie sind in der Lage, Protokollmechanismen quantitativ zu evaluieren (was auch fachübergreifend einsetzbar ist).	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	

Mobilkommunikation: Vorlesung mit Folien und Tafelanschrieb; begleitende Übungen u.a. mit Programmieraufgaben zu einfachen Simulationen drahtloser Systeme.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Mobilkommunikation: Foliensatz; einzelne Kapitel div. Standardlehrbücher (J. Schiller, Mobile Communication, Addison Wesley, 2nd edition; D. Tse und P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, 2005).
<b>Sonstige Hinweise</b>
keine

### 3.36 Wahlpflichtmodul: Model Checking

Modulname	Model Checking / Model Checking
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Model Checking : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Model Checking: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Model Checking: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Model Checking: Logic, imperative Programmierung, Parallelität	
<b>Inhalte</b>	
Model Checking: In dieser Vorlesung werden automatische Verfahren zur Verifikation von Software- oder Hardwaresystemen vorgestellt, d.h. Verfahren, die prüfen, ob ein System korrekt gegenüber spezifizierten Anforderungen ist. Für die Beschreibung der Anforderungen werden dabei temporale Logiken (CTL und LTL) genutzt. Solche Verifikationsverfahren sind in Werkzeugen umgesetzt, von denen eines in der Vorlesung erklärt und in den Übungen genutzt wird.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage für Software oder Hardwaresysteme Anforderungen formal in temporaler Logik zu spezifizieren und kennen Werkzeuge, um diese Anforderungen automatisch zu prüfen. Sie kennen die Funktionsweise von Modelchecking-Algorithmen, und die Unterschiede zwischen linear-time und branching-time Logiken. Sie können neue Forschungsansätze im Bereich Verifikation verstehen und in Bestehendes einordnen. Die Studierenden können einfache formale Beweise selber entwickeln sowie die in der Vorlesung vorgestellten Beweise erklären.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	
Model Checking: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. In praktischen Übungen mit dem Modelcheck-	

er Spin werden die erlernten Kenntnisse angewendet.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Model Checking: Christel Baier, Joost-Pieter Katoen: Principles of Model Checking Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.37 Wahlpflichtmodul: Model-Based Interface Development

Modulname	Model-Based Interface Development / Model-Based Interface Development
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Modellbasierte Entwicklung von Benutzungsschnittstellen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Modellbasierte Entwicklung von Benutzungsschnittstellen: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Modellbasierte Entwicklung von Benutzungsschnittstellen: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Modellbasierte Entwicklung von Benutzungsschnittstellen: Die Vorlesung setzt keine besonderen Vorkenntnisse, die über das vorangegangene Bachelorstudium hinausgehen, voraus.	
<b>Inhalte</b>	
<p>Modellbasierte Entwicklung von Benutzungsschnittstellen: Die Vorlesung beginnt mit einer klärenden Einführung des Modellbegriffs innerhalb der Informatik. Modelle werden hier wie folgt definiert: Ein Modell ist ein Artefakt, wie zum Beispiel eine Zeichnung oder ein physisches Objekt, das einen bestimmten wichtigen Aspekt eines realen Systems wiedergibt oder repräsentiert. Ausgehend von dieser Definition werden eine Reihe von Typen von Artefakten zur Abdeckung verschiedener Aspekte betrachtet. Es beginnt mit Ansätzen zur Aufgabenmodellierung, wo wir verschiedene Methoden und Werkzeuge - mit denen auch gearbeitet wird - betrachten. Danach behandeln wir Dialogmodelle, die in zwei grundverschiedene Gruppen aufgeteilt werden: Kontrollmodelle und Benutzerinteraktionsmodelle. Beide Typen beschreiben eigentlich dieselbe Sache", haben aber verschiedene Ziele und führen zu verschiedenen Ergebnissen. Wir behandeln pro Typ mehrere Ansätze und ihre Querbezüge, wann immer möglich mit Betrachtung entsprechender Werkzeuge. Betrachtet wird auch ein typischer modellbasierter Benutzungsschnittstellen-Entwicklungsprozess. Die gefundenen Erkenntnisse werden anschließend auf den Bereich der Webmodellierung übertragen. Die Probleme sind im wesentlichen die gleichen wie bei der klassischen Benutzungsschnittstelle, allerdings werden andere, dem Bereich angepassten, Modellierungsansätze verwendet. Wir betrachten Problembereichsmodell, Navigationsmodell und Präsentationsmodell für Webauftritte. Neben (noch) rein akademisch interessanten Ansätzen wird intensiv die existierende, lauffähige Webmodellierungsumgebung WebRatio behandelt und praktisch eingesetzt.</p>	

<b>Lernergebnisse / Fachkompetenzen</b>
Faktenwissen: Die Vorlesung vermittelt explizit Kalküle, die beim Modellieren zum Einsatz kommen. Dadurch wird auch deutlich gemacht, welche inhaltlichen Aspekte, die bei der Entwicklung von Benutzungsschnittstellen eine Rolle spielen, mit welchem Modell und auf welcher Abstraktionsebene behandelt werden können. Es werden sowohl die Grundlagen für diese Betrachtungen behandelt, als auch spezielle, für das Web relevante Aspekte und Modellierungskonzepte. Methodisches Wissen: Die Studierenden lernen die behandelten Modellierungstechniken einzusetzen. Es werden teils akademische, teils kommerzielle Modellierungswerkzeuge vorgestellt und ihr Gebrauch in den Übungen erprobt. Transferkompetenz: Die Studierenden lernen wie formale Kalküle in realen Entwurfsszenarien hilfreich eingesetzt werden können. Normativ-bewertend: Die Studierenden lernen den Nutzen, aber auch die Defizite der konkreten Modellierungsansätze, aber auch des modellbasierten Ansatzes an sich kennen. Sie können im Ergebnis bei der Lösung praktischer Aufgaben beurteilen, ob der Einsatz einer modellbasierten Vorgehensweise angebracht ist und welche Vorteile und Risiken damit verbunden sind.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Modellbasierte Entwicklung von Benutzungsschnittstellen: Die Vorlesung wird klassisch mit Beamerpräsentation gehalten und intensiv durch die E-Learning-Umgebung koaLA der Universität Paderborn unterstützt. Hier werden vor der Vorlesung die Folien veröffentlicht, schriftliche Übungsaufgaben gestellt, Software (Modellierungswerkzeuge) und Videoaufzeichnungen aller Vorlesungen zur Verfügung gestellt. Im Rahmen der Vorlesung selbst findet immer wieder an geeigneter Stelle interaktive Gruppenarbeit statt – etwa zum Modellieren mit WebRatio oder der Erstellung komplexer Aufgabenmodelle. In den Übungsgruppen stellen Hörer die von ihnen erarbeiteten Lösungen vor und zur Diskussion. Dazu müssen die Vortragenden ihre Lösung anhand von geeigneten Vortragsfolien präsentieren.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>

Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gerd Szwillus
<b>Lernmaterialien, Literaturangaben</b>
<p>Modellbasierte Entwicklung von Benutzungsschnittstellen:</p> <ul style="list-style-type: none"> <li>• Vorlesungsfolien als PDF zum Herunterladen</li> <li>• Schriftliche Hausaufgaben</li> <li>• Diverse, während der Vorlesung eingesetzte Modellierungswerkzeuge</li> <li>• Voller Zugang zu WebRatio</li> <li>• Ben Shneiderman: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2009</li> <li>• David Benyon: Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design, 2009</li> <li>• Stefano Ceri et al: Designing Data-Intensive Web Applications, 2003</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.38 Wahlpflichtmodul: Network Simulation

Modulname	Network Simulation / Network Simulation
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Network Simulation : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Network Simulation: Vorlesung ( 30h / 105h / EN / SS / 20 ) Network Simulation: Übung ( 45h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Network Simulation: Systemsoftware und systemnahe Programmierung	
<b>Inhalte</b>	
<p>Network Simulation: Der Kurs fokussiert auf dem Simulationswerkzeug OMNeT++. Simulation ist eine der grundlegenden Möglichkeiten (neben Experimenten und mathematischer Analyse), die Leistung von Systemen zu bewerten, auch wenn diese noch nicht in als reales System verfügbar sind.</p> <p>Nach eine grundlegenden Einführung in Simulatio und Modellierung werden anhand von kleinen Beispielprojekten die Möglichkeiten von OMNeT++ exploriert. Letztendlich soll aber ein forschungsrelevantes Projekt in Gruppenarbeit (2-3 Studierende pro Gruppe) durchgeführt werden. Die Themen kommen aus den Gebieten Fahrzeugkommunikation und Sensornetze.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Lernziel ist es, grundlegende Konzepte der Netzwerksimulation zu studieren. Die Studierenden verstehen diese Konzepte und sind in der Lage, dieses Wissen anzuwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Network Simulation: Vorlesung mit praktischen Übungen	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	



Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Network Simulation: Folien, Lehrbücher, Papiere
<b>Sonstige Hinweise</b>
keine

### 3.39 Wahlpflichtmodul: Networked Embedded Systems

Modulname	Networked Embedded Systems / Networked Embedded Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Networked Embedded Systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Networked Embedded Systems: Vorlesung ( 45h / 105h / EN / WS / 60 ) Networked Embedded Systems: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Networked Embedded Systems: Systemsoftware und systemnahe Programmierung	
<b>Inhalte</b>	
Networked Embedded Systems: Ziel des Kurses ist es, vertiefte Einblicke in den Entwurf und die Programmierung eingebetteter Systeme zu erlangen. Der Fokus liegt klar auf der Anwendungsdomäne Sensornetze. Daher werden fundamentale Grundlagen von Sensornetzen untersucht und im Rahmen der Übungen vertieft.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Ziel ist es, grundlegende Konzepte vernetzter eingebetteter Systeme zu verstehen. Die Studierenden sind in der Lage, diese Konzepte anzuwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Networked Embedded Systems: Vorlesung mit praktischen Übungen	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.	

<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Networked Embedded Systems: Folien, Lehrbücher, Papiere
<b>Sonstige Hinweise</b>
keine

### 3.40 Wahlpflichtmodul: Planning and Heuristic Search

Modulname	Planning and Heuristic Search / Planning and Heuristic Search
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Planen und Heuristische Suche : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Planen und Heuristische Suche: Vorlesung ( 45h / 105h / EN / WS / 40 ) Planen und Heuristische Suche: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Planen und Heuristische Suche: Grundlagen symbolischer Modellierung: Funktionen, Relationen, logische Formeln; Design und Analyse von Algorithmen; Grundlagen zur Komplexitätstheorie: Komplexitätsklassen, Reduzierbarkeit, Vollständigkeit.	
<b>Inhalte</b>	
Planen und Heuristische Suche: Die Veranstaltung Planen und Heuristische Suche stellt zwei Ansätze zum Lösen wissensintensiver Aufgaben vor. Im Bereich Planen werden Repräsentationen von Aufgabenstellungen als Planungsproblem in Zustandsräumen oder Planräumen vorgestellt und passende Verfahren diskutiert und analysiert. Im Bereich Heuristische Suche wird das Konzept des Zustandsraums verallgemeinert und ein Programmrahmen für systematische Suchverfahren beschrieben, der es erlaubt, die Suche durch Nutzung heuristischer Informationen über die Problemdomäne zu fokussieren. Als ein Anwendungsbeispiel werden Planungsverfahren als heuristische Suche implementiert. In beiden Bereichen werden theoretische Ergebnisse vorgestellt und bewiesen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage, <ul style="list-style-type: none"> <li>• die Konzepte zur Modellierung von Planungs- und Suchaufgaben zu benennen und zu erklären,</li> <li>• die Vorgehensweisen von verschiedenen Planungsverfahren sowie von heuristischen Suchverfahren zu beschreiben,</li> <li>• einfache Aufgabenstellungen als Planungs- bzw. Suchaufgaben zu erkennen und zu repräsentieren,</li> <li>• darin Ansätze zur Erstellung brauchbarer Heuristiken zu entdecken,</li> <li>• theoretische Ergebnisse als Hinweise für die Auswahl von Modellierung und Verfahren zu verwenden.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> </ul>	

<ul style="list-style-type: none"> <li>• Lernmotivation</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Planen und Heuristische Suche: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Übungen</li> <li>• Diskussion</li> <li>• Hausaufgaben</li> <li>• Referenzimplementationen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Theodor Lettmann
<b>Lernmaterialien, Literaturangaben</b>
Planen und Heuristische Suche: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Lehrbücher: 1.) J. Pearl: Heuristics, Addison-Wesley (1984), 2.) S.J. Russell, P. Norvig: Artificial Intelligence: A Modern Approach, Prentice Hall, 1995, 3.) S. Edelkamp, S. Schrödl: Heuristic Search: Theory and Applications, Elsevier, 2012, 4.) M. Ghallab, D. Nau, P. Traverso: Automated Planning,</li> </ul>

Morgan Kaufmann, 2004

- Übungsaufgaben
- Liste von klassischen und aktuellen Papieren, z.B. R. Eberdt, R. Drechsler: Weighted A\*Search - Unifying View and Application, J. Artificial Intelligende, pp. 1310-1342, 2009 item Online Material, z.B. H. Geffner, B. Bonet: A Concise Introduction to Models and Methods for Automated Planning, doi: 10.2200/S00513ED1V01Y201306AIM022, 2013

**Sonstige Hinweise**

keine

### 3.41 Pflichtmodul: Projektgruppe

Modulname	Projektgruppe / Proje Group
Workload	600 h
Leistungspunkte	20 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Projektgruppe : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Projektgruppe: Vorlesung ( 30h / 540h / EN / WS oder SS / 10 ) Projektgruppe: Besprechungen, Präsentation von Teilergebnissen ( 30h / 0 h / EN / WS oder SS / 10 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Projektgruppe: Abhängig vom Thema.	
<b>Inhalte</b>	
Projektgruppe: In einer Projektgruppe bearbeitet eine Gruppe von in der Regel 8-16 Studierenden über den Zeitraum eines Jahres (zwei Semester) ein vom Veranstalter vorgegebenes Thema. Inhaltlich sollen Projektgruppen die Studierenden an aktuelle Forschungsthemen heranzuführen und durch die Teamarbeit auf die Arbeitsweise der industriellen Praxis vorbereiten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
In der Projektgruppe wird Teamarbeit und Organisation eines Projekts praktisch erprobt und erlernt; hierdurch werden die Teilnehmer auf die spätere industrielle Berufspraxis vorbereitet. Die Studierenden lernen umfangreiche Entwicklungsprozesse im Team aus eigener Anschauung kennen. Durch die ausdrückliche Arbeitsteilung entsteht der Zwang, über eigene Arbeiten innerhalb der Gruppe zu berichten und die Ergebnisse zu vertreten.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
<p>Projektgruppe:</p> <ul style="list-style-type: none"> <li>• Die Zahl der Projektgruppenteilnehmer ist auf 16 Personen begrenzt.</li> <li>• Es finden Plenumstreffen (alle Teilnehmer und der Veranstalter) statt, insbesondere zur Vermittlung gemeinsam erforderlichen Wissens (Seminarphasen zu Beginn der beiden Semester) und zur Planung der weiteren Arbeit.</li> <li>• Es werden feste Verantwortlichkeiten zwischen den Teilnehmern aufgeteilt, die über die gesamte Projektlaufzeit oder auch nur kurzfristig (ad-hoc-Aufgaben) Bestand haben können.</li> <li>• Es werden Untergruppen zu einzelnen Themen gebildet, die selbständig und termingebunden Aufgaben vorantreiben und dem Plenum Rechenschaft ablegen müssen.</li> <li>• Typischerweise erarbeitet jede Projektgruppe auch eine Repräsentation ihrer Arbeit in einer Webseite.</li> <li>• Am Ende jedes der beiden Semester ist ein Bericht zu erstellen, der in jedem Aspekt von den Teilnehmern gestaltet und mit Inhalt gefüllt wird.</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Phasenbezogene Prüfung (100 % der Modulnote)  Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
<p>Studienleistung: Praktikumsarbeit  Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</p>
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 8 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Projektgruppe: Abhängig vom Thema.



**Sonstige Hinweise**

Im Modul Projektgruppe ist die erfolgreiche Bearbeitung von Projekten durch die Abgabe von Software und Dokumentation als phasenbezogene Prüfung nachzuweisen. Es wird eine Note für die Gesamtheit der bearbeiteten Projekte vergeben.

### 3.42 Wahlpflichtmodul: Public-Key Cryptography

Modulname	Public-Key Cryptography / Public-Key Cryptography
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Public-Key Cryptography : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Public-Key Cryptography: Vorlesung ( 45h / 105h / EN / WS / 50 ) Public-Key Cryptography: Übung ( 30h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Public-Key Cryptography: Elementare Grundlagen der Kryptographie, wie sie in einer Vorlesung wie "Einführung in die Kryptographie" vermittelt werden.	
<b>Inhalte</b>	
Public-Key Cryptography: Diese Veranstaltung vermittelt grundlegende Techniken zur Konstruktion und formalen Sicherheitsanalyse von Public-Key Kryptosystemen. Dies umfasst insbesondere digitale Signaturen, Public-Key Verschlüsselung, identitätsbasierte Verschlüsselung und ggfs. weitere verwandte Konzepte. Der Hauptfokus liegt hierbei auf der Vorstellung der Techniken, die modernen Kryptosystemen und deren Sicherheitsbeweisen zugrunde liegen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende kennen etablierte Techniken zur Konstruktion beweisbar sicherer kryptographischer Public-Key Verfahren. Sie sind in der Lage Sicherheitsanforderungen formal zu definieren, kennen die etablierten Standard-Definitionen und ihre Grenzen und sind in der Lage die Beziehungen zwischen diesen Definitionen formal zu analysieren. Sie können Standard-Beweistechniken auch auf neue Kryptosysteme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Public-Key Cryptography: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> </ul>	

<ul style="list-style-type: none"> <li>• Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr., Tibor Jager
<b>Lernmaterialien, Literaturangaben</b>
Public-Key Cryptography: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Tafelanschrieb</li> <li>• Originalliteratur</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.43 Wahlpflichtmodul: Reconfigurable Computing

Modulname	Reconfigurable Computing / Reconfigurable Computing
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Reconfigurable Computing : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Reconfigurable Computing: Vorlesung ( 30h / 105h / EN / WS / 50 ) Reconfigurable Computing: Übung ( 45h / 0h / EN / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Reconfigurable Computing: Kenntnisse aus Digitaltechnik und Rechnerarchitektur sind hilfreich.	
<b>Inhalte</b>	
Reconfigurable Computing: Die Veranstaltung vermittelt Kenntnisse in Architekturen und Entwurfsmethoden für rekonfigurierbare Hardware und stellt Anwendungen im Bereich des Hochleistungsrechnens und der eingebetteten Systeme vor.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• den Aufbau rekonfigurierbarer Hardwarebausteine zu erklären,</li> <li>• die wesentlichen Entwurfsmethoden zu benennen und zu analysieren und</li> <li>• die Eignung rekonfigurierbarer Hardware für verschiedene Einsatzgebiete zu beurteilen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Reconfigurable Computing: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Rechnerübungen mit rekonfigurierbaren Systemen</li> </ul>	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten)	

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Reconfigurable Computing: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• S. Hauck and A. DeHon (editors): Reconfigurable Computing, Volume 1: The Theory and Practice of FPGA-Based Computation, Morgan Kaufmann, 2008</li> <li>• Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.44 Wahlpflichtmodul: Routing and Data Management in Networks

Modulname	Routing and Data Management in Networks / Routing and Data Management in Networks
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Routing and Data Management in Networks : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Routing and Data Management in Networks: Vorlesung ( 45h / 105h / EN / SS / 40 ) Routing and Data Management in Networks: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Routing and Data Management in Networks: Algorithmen-Entwurf, theoretische Korrektheit und Effizienzbeweise, Werkzeuge aus der Kombinatorik und Wahrscheinlichkeitstheorie.	
<b>Inhalte</b>	
Routing and Data Management in Networks: Routing und Datenmanagement sind grundlegenden zu lösende Aufgaben, um eine effiziente Verwendung von großen Netzwerken wie z.B. dem Internet, Peer-to-Peer-Systemen, oder drahtlosen mobilen Ad-hoc-Netzwerke zu ermöglichen. Diese Vorlesung befasst sich mit Algorithmen und deren Analyse für das Routing und Datenmanagement in solchen Systemen und beschreibt insbesondere Methoden für den Umgang mit ihrer Dynamik (Bewegung von Knoten, Beitritt und Austritt von Knoten). Dabei werden insbesondere lokale, verteilte Algorithmen, häufig als online Algorithmen betrachtet.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden lernen die wichtigsten Techniken im Bereich des Routing und Datenmanagements von großen Netzwerken kennen. Sie können entscheiden, in welcher Situation welcher Datenmanagement-, Scheduling- oder Routing-Algorithmus geeignet ist. Sie können Algorithmen an neue Situationen anpassen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
Routing and Data Management in Networks: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Bearbeitung der Hausaufgaben, Mitarbeit in den Übungen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Friedhelm Meyer auf der Heide
<b>Lernmaterialien, Literaturangaben</b>
Routing and Data Management in Networks: Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Frank Thomson Leighton, M. Kaufmann Publishers, 1992. Originalarbeiten, Skript, Foliensatz der Vorlesung, Übungsblätter
<b>Sonstige Hinweise</b>
keine

### 3.45 Pflichtmodul: Seminar I

Modulname	Seminar I / Seminar I
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Seminar : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Seminar: Seminar ( 30h / 120 h / EN / WS oder SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Seminare aus dem Masterstudiengang Informatik.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Seminar: Abhängig vom Seminarthema.	
<b>Inhalte</b>	
<p>Seminar: In Seminaren erarbeiten sich die Teilnehmer ein Thema, welches in einem Vortrag mit anschließender Diskussion und einer schriftlichen Ausarbeitung präsentiert wird. Vortragsmaterial und Ausarbeitung dienen dabei unterschiedlichen Zielen: Während das Vortragsmaterial zur Unterstützung des Vortrags dient (der in engen zeitlichen Grenzen abläuft), dient die Ausarbeitung dazu, sich zu einem späteren Zeitpunkt detailliert über das Thema informieren zu können.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Im Seminar sollen die Studierenden Techniken zur selbständigen Erarbeitung von nichttrivialem Stoff erlernen, indem sie sich in ein forschungsnahes Teilgebiet der Informatik einarbeiten. Sie sollen lernen, einen Vortrag zu planen, der sich an zeitliche Vorgaben (üblicherweise 45 bis 60 Minuten) hält, und dabei inhaltliche Prioritäten zu setzen. Die Teilnehmer sollen praktisch erfahren, wie man als Zuschauer aus einem Vortrag Kenntnisse aufnimmt, und in Diskussionen Meinungen und Information austauschen. Seminare dienen auch der Vermittlung rhetorischer Fähigkeiten bei Vortrag und Diskussion. Die Teilnehmer sollen lernen, den Vortrag entlang einer inhaltlichen Linie zu strukturieren und verschiedene Mittel zur Illustration komplexer Sachverhalte zu nutzen. Ebenso soll der angemessene Umgang mit Literatur gelernt werden.</p>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Medienkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	



<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Seminar: Referate mit schriftlicher Ausarbeitung und Vortrag.
<b>Prüfungsleistung (Dauer)</b>
Seminarvortrag (45-60 Minuten) und schriftliche Ausarbeitung Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Seminar: Abhängig vom Seminarthema.
<b>Sonstige Hinweise</b>
keine

### 3.46 Pflichtmodul: Seminar II

Modulname	Seminar II / Seminar II
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Seminar : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Seminar: Seminar ( 30h / 120 h / EN / WS oder SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Seminare aus dem Masterstudiengang Informatik.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Seminar: Abhängig vom Seminarthema.	
<b>Inhalte</b>	
<p>Seminar: In Seminaren erarbeiten sich die Teilnehmer ein Thema, welches in einem Vortrag mit anschließender Diskussion und einer schriftlichen Ausarbeitung präsentiert wird. Vortragsmaterial und Ausarbeitung dienen dabei unterschiedlichen Zielen: Während das Vortragsmaterial zur Unterstützung des Vortrags dient (der in engen zeitlichen Grenzen abläuft), dient die Ausarbeitung dazu, sich zu einem späteren Zeitpunkt detailliert über das Thema informieren zu können.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Im Seminar sollen die Studierenden Techniken zur selbständigen Erarbeitung von nichttrivialem Stoff erlernen, indem sie sich in ein forschungsnahes Teilgebiet der Informatik einarbeiten. Sie sollen lernen, einen Vortrag zu planen, der sich an zeitliche Vorgaben (üblicherweise 45 bis 60 Minuten) hält, und dabei inhaltliche Prioritäten zu setzen. Die Teilnehmer sollen praktisch erfahren, wie man als Zuschauer aus einem Vortrag Kenntnisse aufnimmt, und in Diskussionen Meinungen und Information austauschen. Seminare dienen auch der Vermittlung rhetorischer Fähigkeiten bei Vortrag und Diskussion. Die Teilnehmer sollen lernen, den Vortrag entlang einer inhaltlichen Linie zu strukturieren und verschiedene Mittel zur Illustration komplexer Sachverhalte zu nutzen. Ebenso soll der angemessene Umgang mit Literatur gelernt werden.</p>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Medienkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Seminar: Referate mit schriftlicher Ausarbeitung und Vortrag.
<b>Prüfungsleistung (Dauer)</b>
Seminarvortrag (45-60 Minuten) und schriftliche Ausarbeitung Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
keine
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Seminar: Abhängig vom Seminarthema.
<b>Sonstige Hinweise</b>
keine

### 3.47 Wahlpflichtmodul: Software Analysis

Modulname	Software Analysis / Software Analysis
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Software Analysis : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Software Analysis: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Software Analysis: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Software Analysis: Logic, imperative programming	
<b>Inhalte</b>	
Software Analysis: In der Vorlesung werden statische Analysemethoden für Programm vorgestellt. Die Information, die dabei gewonnen wird, kann für die Optimierungsphase von Compilern, aber auch für den Nachweis der Korrektheit von Programmen genutzt werden.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden kennen die wesentlichen Prinzipien von Datenflussanalysen und ihre Vor- und Nachteile. Sie können einschätzen, wann Datenflussanalysen sinnvoll und wann pfadsensitive Analysen benötigt sind. Studierende können eigene Analysen entwerfen und wissen, wie diese umzusetzen sind. Sie kennen das Prinzip der Überapproximation.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	
Software Analysis: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei	

Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Software Analysis: Nielson, Nielson, Hankin: Principles of Program Analysis Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.48 Wahlpflichtmodul: Software Quality Assurance

Modulname	Software Quality Assurance / Software Quality Assurance
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Software Quality Assurance : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Software Quality Assurance: Vorlesung ( 45h / 105h / EN / SS / 90 ) Software Quality Assurance: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Software Quality Assurance: Programmierung, Modellierung, Modellbasierte Softwareentwicklung	
<b>Inhalte</b>	
Software Quality Assurance: Das Ziel der Vorlesung ist die Behandlung von Ansätzen, Technologien und Strategien für die Qualitätssicherung von Softwaresystemen. Dies beinhaltet einerseits konstruktive Ansätze wie Design Pattern, Anti-Pattern, domänenspezifische Sprachen, modellgetriebene Softwareentwicklung, Qualitätsmodelle und Architekturstile und andererseits analytische Ansätze wie statische Reviewtechniken und dynamisch Testtechniken. Des Weiteren werden Ansätze für die Verbesserung des Softwareentwicklungsprozesses und internationale Standards wie ISO 9001, 9126, CMM, usw. behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind in der Lage, Qualitätseigenschaften von Entwicklungsprozessen, Softwaremodellen bzw. -systemen zu benennen. Sie kennen Techniken zur konstruktiven bzw. analytischen Sicherstellung von Qualitätseigenschaften und können diese geeignet einsetzen. Sie kennen die wesentlichen Standards für die Bewertung von Prozess- und Produktqualitäten. Sie können ausgewählte, aktuelle Forschungsansätze im Bereich Prozess- und Softwarequalitätssicherung verstehen und einordnen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Empathie</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	

<b>Methodische Umsetzung</b>
Software Quality Assurance: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. In praktischen Übungen insbesondere mit Testwerkzeugen werden die erlernten Kenntnisse angewendet.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gregor Engels
<b>Lernmaterialien, Literaturangaben</b>
Software Quality Assurance: Daniel Galin: Software Quality Assurance: From Theory to Implementation, Pearson / Addison Wesley, 2004 Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.49 Pflichtmodul: Studium Generale

Modulname	Studium Generale / General Studies
Workload	360 h
Leistungspunkte	12 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Studium Generale : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Studium Generale: Vorlesung ( 90h / 225h / DE / WS / 30 ) Studium Generale: Übung ( 45h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Beliebige Veranstaltungen außerhalb der Informatik können gewählt werden.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Inhalte</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden erweitern ihren wissenschaftlichen Horizont über die Grenzen der Informatik und des gewählten Nebenfaches hinaus. Je nach gewählter Veranstaltung haben sie Kompetenzen im Bereich Kommunikationsfähigkeit, Teamarbeit und Präsentationstechniken erworben.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Medienkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	
<b>Methodische Umsetzung</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Prüfungsleistung (Dauer)</b>	
Prüfung im Studium Generale Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.	



<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Qualifizierte Teilnahme: Qualifizierte Teilnahme im Studium Generale Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
keine
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 4 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dozenten der Universität Paderborn
<b>Lernmaterialien, Literaturangaben</b>
Studium Generale: Abhängig von den gewählten Veranstaltungen.
<b>Sonstige Hinweise</b>
Ist kein Nebenfach gewählt, muss eine beliebige Kombination von Veranstaltungen außerhalb der Informatik und im Umfang von 12 LP muss gewählt werden. Die angegebene Verteilung der LP auf Lehrveranstaltungen ist nur exemplarisch.

### 3.50 Wahlpflichtmodul: Type Systems for Correctness and Security

Modulname	Type Systems for Correctness and Security / Type Systems for Correctness and Security
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Typsysteme für Korrektheit und Sicherheit : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Typsysteme für Korrektheit und Sicherheit: Vorlesung ( 45h / 105h / EN / WS / 30 ) Typsysteme für Korrektheit und Sicherheit: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Typsysteme für Korrektheit und Sicherheit: Aus der Vorlesung “Grundlagen der Programmiersprachen”: Spracheigenschaften, Syntaktische Strukturen, Datentypen, Funktionale Programmierung	
<b>Inhalte</b>	
<p>Typsysteme für Korrektheit und Sicherheit: Typsysteme in Programmiersprache helfen fehlerhaftes Verhalten von Beginn an zu Vermeiden. Sie erlauben wertvolle Rückmeldungen für Entwickler, um Fehler und Systemabstürze zu vermeiden oder sogar Sicherheitslücken zu erkennen. In dieser Vorlesung werden wir Typsysteme entwickeln und untersuchen. Wir werden die Theorie behandeln, die Eigenschaften mit denen Typsysteme uns in der Softwareentwicklung unterstützen besprechen und die Implementierung aktueller Typsysteme untersuchen.</p> <p>Wir werden einen pragmatischen Ansatz durchführen und im Laufe der Vorlesung und im Rahmen von Übungen die Implementierung von Typcheckern erüben. Weiterhin untersuchen wir die Typsysteme von bekannten Programmiersprachen wie Java oder Scala genauer.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Teilnehmer sind in der Lage die Definition und Implementierung von Typsystemen zu verstehen und selbstständig zu bearbeiten. Die erlernten Kenntnisse werden im Kurs diskutiert und vertieft, so dass die Teilnehmer danach in der Lage sind mit dem erworbenen Faktenwissen und der Methodenkompetenz den Vorlesungsstoff auf andere Problemstellungen anwenden zu können.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	

<ul style="list-style-type: none"> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>
<b>Methodische Umsetzung</b>
Typsysteme für Korrektheit und Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Diskussionen</li> <li>• Lesen</li> <li>• Übungen mit begleitender Implementierung</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Ben Hermann
<b>Lernmaterialien, Literaturangaben</b>
Typsysteme für Korrektheit und Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• Übungsaufgaben</li> </ul> Weiterhin wird folgende Literatur empfohlen: <ul style="list-style-type: none"> <li>• Benjamin C. Pierce. 2002. Types and Programming Languages. The MIT Press.</li> <li>• Benjamin C. Pierce. 2004. Advanced Topics in Types and Programming Languages. The MIT Press.</li> </ul>

<b>Sonstige Hinweise</b>
keine

### 3.51 Wahlpflichtmodul: Usability Engineering Practice

Modulname	Usability Engineering Practice / Usability Engineering Practice
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Praxis des Usability Engineering : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Praxis des Usability Engineering: Vorlesung ( 45h / 105h / EN / WS oder SS / 40 ) Praxis des Usability Engineering: Übung ( 30h / 0h / EN / WS oder SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Praxis des Usability Engineering: Kenntnisse aus der Vorlesung „Interaktionsdesign“ des Bachelorstudiengangs Informatik werden vorausgesetzt, da hier die grundlegenden Kenntnisse zu Usability-Methoden erarbeitet werden.	
<b>Inhalte</b>	
Praxis des Usability Engineering: Die Vorlesung „Praxis des Usability Engineering“betzt die Bachelorveranstaltung „Interaktionsgestaltung“ fort. Nachdem dort grundlegende Verfahren des Usability Engineering (Benutzertests, Inspektionsmethoden) eingeführt werden, werden nun einzelne ausgewählte und aktuelle Verfahren (z.B. Cognitive Walkthrough, CardSorting, Value-Centered-Design) vertiefend und in praktischen Übungen ausführlich behandelt. Außerdem betrachtet die Veranstaltung Konzepte und Methoden, die den Begriff des Usability Engineering in vielerlei Hinsicht ausgeweitet betrachten, durch Behandlung etwa von User Experience, Extreme Usability, Ästhetik oder Gesundheits- und Sicherheitsaspekten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Zum einen wird konkretes Faktenwissen zu aktuellen Usabilitymethoden und -ansätzen vermittelt. Methodisch erlernen die Hörer die Anwendung dieser Methoden an hinreichend komplexen, realitätsnahen Beispielen. Damit sind die Hörer am Ende in der Lage die Usability von Softwaresystemen systematisch und mit geeigneter Werkzeugunterstützung zu beurteilen, lernen aber auch die Grenzen solcher Beurteilungsvorgänge kennen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> </ul>	

<b>Methodische Umsetzung</b>
Praxis des Usability Engineering: Die Vorlesung wird klassisch mit Beamerpräsentation gehalten und intensiv durch die E-Learning-Umgebung koaLA der Universität Paderborn unterstützt. Hier werden vor der Vorlesung die Folien veröffentlicht, schriftliche Übungsaufgaben gestellt, Software (Modellierungswerkzeuge) und Videoaufzeichnungen aller Vorlesungen zur Verfügung gestellt. Im Rahmen der Vorlesung selbst findet immer wieder an geeigneter Stelle interaktive Gruppenarbeit statt – etwa zur Ästhetik oder zu Modellkonzepten wie dem Value-Centered-Design-Ansatz. In den Übungsgruppen stellen Hörer die von ihnen erarbeiteten Lösungen vor und zur Diskussion. Dazu müssen die Vortragenden ihre Lösung anhand von geeigneten Vortragsfolien präsentieren.
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gerd Szwillus
<b>Lernmaterialien, Literaturangaben</b>
Praxis des Usability Engineering: <ul style="list-style-type: none"> <li>• Vorlesungsfolien als PDF zum Herunterladen</li> <li>• Schriftliche Hausaufgaben</li> <li>• Diverse, während der Vorlesung eingesetzte Modellierungswerkzeuge</li> <li>• Ben Shneiderman: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2009</li> <li>• David Benyon: Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction</li> </ul>

Design, 2009

- Jakob Nielsen und Raluca Budiu: Mobile Usability, 2012
- Effie Lai-Chong Law et al. (ed): Maturing Usability: Quality in Software, Interaction and Value, 2007

**Sonstige Hinweise**

keine

### 3.52 Wahlpflichtmodul: Vehicular Networking

Modulname	Vehicular Networking / Vehicular Networking
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Vehicular Networking : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Vehicular Networking: Vorlesung ( 45h / 105h / EN / SS / 40 ) Vehicular Networking: Übung ( 30h / 0h / EN / SS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Vehicular Networking: Systemsoftware und systemnahe Programmierung	
<b>Inhalte</b>	
Vehicular Networking: Dieser Kurs behandelt wichtige Aspekte sowohl von modernen Vernetzungskonzepten im Auto als auch zwischen Fahrzeugen. Dabei werden Aspekte wie Electronic Control Units, Bussysteme, Fahrerassistenzsysteme, aber auch Verkehrsinformationssysteme, Konzepte der drahtlosen Kommunikation bis hin zu Sicherheit und Privatsphäre behandelt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Ziel ist es, grundlegende Konzepte vernetzter eingebetteter Systeme zu verstehen. Die Studierenden sind in der Lage, diese Konzepte anzuwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Vehicular Networking: Vorlesung mit praktischen Übungen	
<b>Prüfungsleistung (Dauer)</b>	
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.	



<b>Modulteilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Vehicular Networking: Folien, Lehrbücher, Papiere
<b>Sonstige Hinweise</b>
keine

### 3.53 Wahlpflichtmodul: VLSI Testing

Modulname	VLSI Testing / VLSI Testing
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• VLSI Testing : 1</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
VLSI Testing: Vorlesung ( 30h / 120h / EN / WS / 40 ) VLSI Testing: Übung ( 30h / 0h / EN / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
VLSI Testing: Digitaltechnik	
<b>Inhalte</b>	
VLSI Testing: Die Lehrveranstaltung behandelt systematische Verfahren zur Erkennung von Hardware-Defekten in mikroelektronischen Schaltungen. Es werden sowohl Algorithmen zur Erzeugung und Auswertung von Testdaten als auch Hardwarestrukturen zur Verbesserung der Testbarkeit und für den eingebauten Selbsttest vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach Absolvieren des Moduls in der Lage, <ul style="list-style-type: none"> <li>• Fehlermodelle, Maßnahmen zur Verbesserung der Testbarkeit und Werkzeuge zur Unterstützung des Tests zu beschreiben,</li> <li>• die grundlegenden Modelle und Algorithmen für Fehlersimulation und Test zu erklären und anzuwenden, sowie</li> <li>• Systeme im Hinblick auf ihre Testbarkeit zu analysieren und geeignete Teststrategien auszuwählen.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
VLSI Testing: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafel</li> <li>• Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen,</li> </ul>	

Präsentation der Lösungen durch Übungsteilnehmer <ul style="list-style-type: none"> <li>• Praktische Übungen mit verschiedenen Software-Werkzeugen am Rechner</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Mündliche Prüfung (ca. 40 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Moduleilprüfungen</b>
keine
<b>Studienleistung / qualifizierte Teilnahme</b>
Studienleistung: schriftliche Übungsaufgaben Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.
<b>Voraussetzungen für die Teilnahme an der Prüfung</b>
Bestehen der Studienleistung.
<b>Voraussetzungen für die Vergabe von Credits</b>
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
<b>Gewichtung für die Gesamtnote</b>
Das Modul wird mit 6 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Sybille Hellebrand
<b>Lernmaterialien, Literaturangaben</b>
VLSI Testing: <ul style="list-style-type: none"> <li>• Vorlesungsfolien</li> <li>• M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing for Digital, Memory &amp; Mixed-Signal VLSI Circuits, Boston, Dordrecht, London: Kluwer Academic Publishers, 2000</li> <li>• L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann Series in Systems on Silicon, ISBN: 0123705975</li> <li>• Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien im koala-Kurs</li> </ul>
<b>Sonstige Hinweise</b>
keine



# Anhang A

## Überblickstabellen

### A.1 Studienrichtungen und Module

	Algorithm Design (S. 11)	Computer Systems (S. 12)	Intelligence and Data (S. 13)	Networks and Communication (S. 14)	Software Engineering (S. 15)
Adaptive Hardware and Systems (S. 18)	-	X	-	-	-
Advanced Algorithms (S. 21)	X	-	-	-	-
Advanced Compiler Construction (S. 23)	-	-	-	-	X
Advanced Complexity Theory (S. 25)	X	-	-	-	-
Advanced Computer Architecture (S. 28)	-	X	-	-	-
Advanced Distributed Algorithms and Data Structures (S. 30)	X	-	-	X	-
Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 32)	-	-	-	-	X
Algorithmic Game Theory (S. 34)	X	-	-	-	-
Algorithms for Highly Complex Virtual Scenes (S. 36)	X	-	-	-	-
Algorithms for Synthesis and Optimization of Integrated Circuits (S. 39)	-	X	-	-	-
Architektur paralleler Rechnersysteme (S. 41)	-	X	-	-	-
Bitcoins, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 44)	-	-	-	X	-
Build It, Break It, Fix It (S. 47)	-	-	-	-	X
Clustering Algorithms (S. 50)	X	-	X	-	-
Compiler Construction (S. 52)	-	-	-	-	X
Computational Geometry (S. 54)	X	-	-	-	-

Designing code analyses for large-scale software systems (S. 58)	-	-	-	-	X
Empiric performance evaluation (S. 61)	-	X	-	X	X
Foundations of Cryptography (S. 63)	X	-	-	-	-
Fundamentals of Model-Driven Engineering (S. 65)	-	-	-	-	X
Future Internet (S. 68)	-	-	-	X	-
Hardware/Software Codesign (S. 70)	-	X	-	-	-
High-Performance Computing (S. 72)	-	X	-	-	X
Intelligence in Embedded Systems (S. 74)	-	X	X	-	-
Interactive Data Visualization (S. 76)	-	-	X	-	-
Kontextuelle Informatik (S. 79)	-	-	-	-	X
Language-Based Security (S. 82)	-	-	-	-	X
Linear and Integer Optimization (S. 84)	X	-	-	-	-
Logic and Automated Reasoning (S. 86)	-	-	X	-	-
Logic Programming for Artificial Intelligence (S. 89)	-	-	X	-	X
Machine Learning I (S. 92)	-	-	X	-	-
Machine Learning II (S. 94)	-	-	X	-	-
Master-Abschlussarbeit (S. 96)	-	-	-	-	-
Mobile Communication (S. 99)	-	-	-	X	-
Model Checking (S. 101)	-	-	-	-	X
Model-Based Interface Development (S. 103)	-	-	-	-	X
Network Simulation (S. 106)	-	-	-	X	-
Networked Embedded Systems (S. 108)	-	-	-	X	-
Planning and Heuristic Search (S. 110)	-	-	X	-	-
Projektgruppe (S. 113)	-	-	-	-	-
Public-Key Cryptography (S. 116)	X	-	-	-	-
Reconfigurable Computing (S. 118)	-	X	-	-	-
Routing and Data Management in Networks (S. 120)	X	-	-	X	-
Seminar I (S. 122)	-	-	-	-	-
Seminar II (S. 124)	-	-	-	-	-
Software Analysis (S. 126)	-	-	-	-	X
Software Quality Assurance (S. 128)	-	-	-	-	X
Studium Generale (S. 130)	-	-	-	-	-
Type Systems for Correctness and Security (S. 132)	-	-	-	-	X

Usability Engineering Practice (S. 135)	-	-	-	-	X
Vehicular Networking (S. 138)	-	-	-	X	-
VLSI Testing (S. 140)	-	X	-	-	-

## A.2 Module und Lehrveranstaltungen

Adaptive Hardware and Systems (S. 18)		Adaptive Hardware and Systems (S. 77)	
Advanced Algorithms (S. 21)		Advanced Algorithms (S. 77)	
Advanced Compiler Construction (S. 23)		Advanced Compiler Construction (S. 77)	
Advanced Complexity Theory (S. 23)		Advanced Complexity Theory (S. 77)	
Advanced Computer Architecture (S. 28)		Advanced Computer Architecture (S. 77)	
Advanced Distributed Algorithms and Data Structures (S. 30)		Advanced Distributed Algorithms and Data Structures (S. 77)	
Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 32)		Advanced Software Engineering: Methods, Architectures, Industrial Applications (S. 77)	
Algorithmic Game Theory (S. 34)		Algorithmic Game Theory (S. 77)	
Algorithms for High Complex Virtual Scenes (S. 36)		Algorithmen für hochkomplexe virtuelle Szenen (S. 77)	
Algorithms for Synthesis and Optimization of Integrated Circuits (S. 39)		Algorithms for Synthesis and Optimization of Integrated Circuits (S. 77)	
Architektur paralleler Rechensysteme (S. 41)		Architektur paralleler Rechensysteme (S. 77)	
Bitcoin, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 44)		Bitcoin, Cryptocurrencies, and Privacy-Enhancing Technologies (S. 77)	
Build It, Break It, Fix It (S. 47)		Build It, Break It, Fix It (S. 77)	
Clustering Algorithms (S. 50)		Clustering Algorithmen (S. 77)	
Compiler Construction (S. 52)		Compilerbau (S. 77)	
Computational Geometry (S. 54)		Algorithmische Geometrie (S. 77)	
Deductive Verification (S. 56)		Deductive Verification (S. 77)	
Designing code analyses for large-scale software systems (S. 59)		Designing code analyses for large-scale software systems (S. 77)	
Empirical performance evaluation (S. 61)		Empirische Leistungsbewertung (S. 77)	
Foundations of Cryptography (S. 63)		Foundations of Cryptography (S. 77)	
Fundamentals of Model-Driven Engineering (S. 65)		Fundamentals of Model-Driven Engineering (S. 77)	
Future Internet (S. 68)		Future Internet (S. 77)	
Hardware/Software Codesign (S. 70)		Hardware/Software Codesign (S. 77)	
High-Performance Computing (S. 72)		High-Performance Computing (S. 77)	
Intelligence in Embedded Systems (S. 74)		Intelligenz in eingebetteten Systemen (S. 77)	
Interactive Data Visualization (S. 76)		Interaktive Datenvisualisierung (S. 77)	
Kontextuelle Informatik (S. 79)		Kontextuelle Informatik (S. 77)	
Language-Based Security (S. 82)		Language-Based Security (S. 77)	
Linear and Integer Optimization (S. 84)		Linear and Integer Optimization (S. 77)	
Logic- and Automated Reasoning (S. 86)		Logic Programming for Artificial Intelligence (S. 77)	
Logic Programming for Artificial Intelligence (S. 88)		Logik und automatisches Beweisen (S. 77)	
Machine Learning I (S. 90)		Maschinelles Lernen I (S. 77)	
Machine Learning II (S. 91)		Maschinelles Lernen II (S. 77)	
Master-Abschleissarbeit (S. 96)		Masterarbeit (S. 77)	
Mobile Communication (S. 99)		Mobile Kommunikation (S. 77)	
Model Checking (S. 101)		Model Checking (S. 77)	
Model-Based Interface Development (S. 103)		Modellbasierte Entwicklung von Benutzungsschnittstellen (S. 77)	
Network Simulation (S. 106)		Network Simulation (S. 77)	
Networked Embedded Systems (S. 108)		Networked Embedded Systems (S. 77)	
Planning and Heuristic Search (S. 110)		Planen und Heuristische Suche (S. 77)	
Projektgruppe (S. 113)		Projektgruppe (S. 77)	
Public-Key Cryptography (S. 116)		Public-Key Cryptography (S. 77)	
Reconfigurable Computing (S. 118)		Reconfigurable Computing (S. 77)	
Routing and Data Management in Networks (S. 120)		Routing and Data Management in Networks (S. 77)	
Seminar I (S. 122)		Seminar (S. 77)	
Seminar II (S. 124)		Seminar II (S. 77)	
Software Analysis (S. 126)		Software Analysis (S. 77)	
Software Quality Assurance (S. 128)		Software Quality Assurance (S. 77)	
Studium Generale (S. 130)		Studium Generale (S. 77)	
Type Systems for Correctness and Security (S. 132)		Type Systems for Correctness and Security (S. 77)	
Usability Engineering Practice (S. 135)		Praxis des Usability Engineering (S. 77)	
Vehicle Networking (S. 138)		Vehicle Networking (S. 77)	
VLSI Testing (S. 140)		VLSI Testing (S. 77)	



---

**HERAUSGEBER  
PRÄSIDIUM DER UNIVERSITÄT PADERBORN  
WARBURGER STR. 100  
33098 PADERBORN**

**[HTTP://WWW.UNI-PADERBORN.DE](http://www.uni-paderborn.de)**

---

**ISSN 2199-2819**