

## **AMTLICHE MITTEILUNGEN**

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

**AUSGABE 43.17 VOM 16. JUNI 2017**

---

## **PRÜFUNGSORDNUNG**

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

**VOM 16. JUNI 2017**

## Prüfungsordnung für den Bachelorstudiengang 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:

I. Allgemeines	4
§ 1 Ziele des Studiums und Zweck der Prüfung	4
§ 2 Akademischer Grad	4
§ 3 Studienbeginn	4
§ 4 Zugangsvoraussetzungen	4
§ 5 Regelstudienzeit und Studiumumfang	5
§ 6 Module	6
§ 7 Anerkennung von Leistungen	6
II. Prüfungsorganisation	7
§ 8 Prüfungsausschuss	7
§ 9 Prüfende und Beisitzende	9
III. Bachelorprüfung	9
§ 10 Ziel, Umfang und Art der Bachelorprüfung	9
§ 11 Zulassung zur Bachelorprüfung	11
§ 12 Anmeldung und Prüfungsfristen	12
§ 13 Abschluss eines Moduls	13
§ 14 Prüfungsleistungen in den Modulen	13
§ 15 Formen der Prüfungsleistungserbringung in den Modulen, Studienleistungen und qualifizierte Teilnahme	13
§ 16 Bewertung von Leistungen in den Modulen	15
§ 17 Abschlussarbeit	16
§ 18 Annahme und Bewertung der Bachelorarbeit	17
§ 19 Bewertung der Bachelorprüfung und Bildung der Gesamtnote	17
§ 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 Bachelorurkunde	22
§ 26 Einsicht in die Prüfungsakten	22
IV. Schlussbestimmungen	23
§ 27 Ungültigkeit der Bachelorprüfung	23
§ 28 Aberkennung des Bachelorgrades	23
§ 29 Übergangsbestimmungen	24
§ 30 Inkrafttreten und Veröffentlichung	24
Anhänge	25
Anhang 1: Module, Zulassungsvoraussetzungen und Prüfungsformen und -modalitäten	25
Anhang 2: Nebenfachvereinbarungen für die Standardnebenfächer im Bachelorstudiengang Informatik	30
Anhang 3: Studienverlaufsplan	33
Anhang 4: Modulhandbuch	34

## I. Allgemeines

### § 1

#### Ziele des Studiums und Zweck der Prüfung

- (1) Die Bachelorprüfung bildet einen ersten berufsqualifizierenden Abschluss des Studiums der Informatik. Das Bachelorstudium im Studiengang Informatik gliedert sich in zwei Abschnitte:
  1. Der erste Abschnitt (1. bis 4. Semester, teilweise im 5. Semester) vermittelt die notwendige Grundlage für ein wissenschaftlich fundiertes Informatikstudium.
  2. Der zweite Abschnitt (teilweise im 5., Abschluss im 6. Semester) dient der Vermittlung eines breiten Spektrums an allgemeinem wissenschaftlichem Informatikwissen und schließt mit der Bachelorprüfung ab, die internationalen Standards entspricht. In diesen Abschnitt fällt auch die optionale berufspraktische Tätigkeit. Der zweite Studienabschnitt ist so ausgelegt, dass ein Auslandssemester durchgeführt werden kann.
- (2) Durch die Bachelorprüfung soll festgestellt werden, ob die Studierenden die für die Berufspraxis notwendigen wissenschaftlichen Grundlagen und Fachkenntnisse erworben haben, die Zusammenhänge ihres Faches überblicken und die Fähigkeit besitzen, Probleme der Informatik zu erkennen, zur Lösung eine geeignete wissenschaftliche Methode auszuwählen und sachgerecht anzuwenden.
- (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 im Hinblick auf die Auswirkungen des technologischen Wandels verantwortlich zu handeln.

### § 2

#### Akademischer Grad

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

### § 3

#### Studienbeginn

Der Studienbeginn ist das Wintersemester.

### § 4

#### Zugangsvoraussetzungen

- (1) In den Bachelorstudiengang 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 besitzt oder die Voraussetzungen für in der beruflichen Bildung

Qualifizierte oder die Voraussetzungen der Bildungsausländerhochschulzugangsverordnung erfüllt oder die Voraussetzungen gemäß § 49 Abs. 11 HG nachweist. Im Falle des § 49 Abs. 11 HG sind die studiengangsbezogene besondere fachliche Eignung sowie eine den Anforderungen der Hochschule entsprechende Allgemeinbildung nachzuweisen. Nähere Einzelheiten ergeben sich aus der Ordnung zur Feststellung der besonderen studiengangsbezogenen fachlichen Eignung für die Bachelorstudiengänge Computer Engineering, Elektrotechnik, Informatik, Mathematik und Technomathematik an der Universität Paderborn in der jeweils gültigen Fassung sowie der Rahmenordnung der Universität Paderborn zur Feststellung der Allgemeinbildung auf Hochschulniveau in der jeweils gültigen Fassung.

2. als Studienbewerberin oder Studienbewerber, die ihre bzw. der seine Zugangsvoraussetzungen nicht an einer deutschsprachigen 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.
- (2) Die Einschreibung ist abzulehnen, wenn
1. die in Abs. 1 genannten Voraussetzungen nicht vorliegen oder
  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 Bachelorstudiengang Informatik der Universität Paderborn als auch die endgültig nicht bestandene Prüfung eine erhebliche inhaltliche Nähe zu einer Prüfung eines Pflichtmoduls des Bachelorstudiengangs Informatik der Universität Paderborn aufweisen. Die Feststellung über erhebliche inhaltliche Nähe trifft der Prüfungsausschuss.

## § 5

### Regelstudienzeit und Studienumfang

- (1) Die Regelstudienzeit beträgt einschließlich der Bachelorprüfung sechs Semester. Es wird von einem Gesamtarbeitsaufwand (Workload) für die Studierenden von 5.400 Stunden (= 180 Leistungspunkte) ausgegangen.
- (2) Das Bachelorstudium umfasst Module mit einem Gesamtumfang von 180 Leistungspunkten (LP). Je nach Nebenfach werden dort Module im Umfang von 18 bis 22 LP studiert. Im Hauptfach Informatik werden ausschließlich Pflichtmodule angeboten. Diese enthalten im ersten Studienabschnitt ausschließlich Pflichtmodule (105 LP) und im zweiten Studienabschnitt abgesehen vom Modul Schlüsselqualifikation (5 LP) ausschließlich Wahlpflichtmodule (30 LP). Dazu kommen Module im Studium Generale (maximal 7 LP je nach Nebenfach) und die Abschlussarbeit inklusive Arbeitsplanung (15 LP).
- (3) Die vergebenen LP entsprechen den im Rahmen des European Credit Transfer Systems (ECTS) zu vergebenden Punktzahlen. Ein LP entspricht einem Arbeitsaufwand 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.

- (4) Die Fakultät für Elektrotechnik, Informatik und Mathematik hat auf der Grundlage dieser Prüfungsordnung einen beispielhaften Studienverlaufsplan und Modulbeschreibungen in einem Modulhandbuch erstellt. Diese Unterlagen geben insbesondere Aufschluss über die Ziele der einzelnen Module und der den Modulen zugeordneten Lehrveranstaltungen sowie über die notwendigen Vorkenntnisse und die 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 Instituts für Informatik veröffentlicht.
- (5) Im Modul 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 Bachelorstudiengang Informatik wird in modularisierter Form angeboten. Module sind thematisch und zeitlich abgerundete, in sich abgeschlossene und mit LP versehene, prüfbare eigenständige Qualifikationseinheiten. Die Module setzen sich aus einer oder mehreren Lehrveranstaltungen zusammen, haben einen Umfang von 4 bis 15 LP und können in der Regel innerhalb von ein bis zwei Semestern abgeschlossen werden.
- (2) Ein Modul kann Pflicht- und Wahlpflichtveranstaltungen enthalten. Enthält ein Modul Wahlpflichtveranstaltungen, werden diese aus einem Veranstaltungskatalog gewählt, der Teil der Modulbeschreibung ist.
- (3) Durch die Wahl der Lehrveranstaltung im Modul „Vertiefung“ ist eine Vertiefung in einem der vier Gebiete möglich.

## **§ 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 Unterscheid 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 einstuft.
- (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 Bachelorstudiengang 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. Die Amtszeit der Mitglieder aus der Gruppe der Hochschullehrerinnen und Hochschullehrer und des Mitglieds aus der Gruppe 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 und 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 die Beisitzenden. Prüfende sind alle selbstständig Lehrende der Veranstaltungen, in denen nach Maßgabe des Curriculums und der Modulbeschreibungen Prüfungsleistungen erbracht werden können. Als Beisitzende bzw. Beisitzender kann bestellt werden, wer mindestens die entsprechende Bachelorprüfung oder eine vergleichbare Prüfung abgelegt hat.
- (2) Die Prüfenden sind in ihrer Prüfungstätigkeit unabhängig.
- (3) Die Kandidatin bzw. der Kandidat kann für die Bachelorarbeit 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 Prüfungsausschusses sorgt dafür, dass der Kandidatin bzw. dem Kandidaten 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. Bachelorprüfung

### § 10 Ziel, Umfang und Art der Bachelorprüfung

- (1) Die Bachelorprüfung erstreckt sich auf die folgenden Gebiete:
  1. Softwaretechnik
  2. Algorithmen und Komplexität
  3. Computersysteme
  4. Daten und Wissen
  5. Mathematik
  6. ein Nebenfach nach Wahl der Kandidatin oder des Kandidaten.

Die Gebiete 1 bis 4 heißen im folgenden Informatikgebiete.

- (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, welche im Anhang 2 dieser Ordnung enthalten sind. Die Stundenpläne werden im Zuge der Erstellung des Vorlesungsverzeichnisses koordiniert.

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 Bachelorprüfung besteht aus
1. studienbegleitenden Modulprüfungen des ersten Abschnitts im Hauptfach einschließlich Mathematik über Inhalte von Modulen mit einem Umfang von 105 LP,
  2. studienbegleitenden Modulprüfungen des zweiten Abschnitts im Hauptfach über Inhalte von Modulen mit einem Umfang von 30 LP sowie zum Modul Schlüsselqualifikation mit einem Umfang von 5 LP,
  3. studienbegleitenden Modulprüfungen im Nebenfach über Inhalte von Modulen mit einem Umfang von mindestens 18 bis höchstens 22 LP,
  4. dem Modul Studium Generale, das in Abhängigkeit des Nebenfachs einen Umfang von mindestens 3 und höchstens 7 LP hat und in dem eine Prüfungsleistung zu erbringen ist,
  5. dem Modul Bachelor-Abschlussarbeit (15 LP) einschließlich eines Arbeitsplanes, der Bachelorarbeit und eines Vortrages von etwa 30 Minuten Dauer.
- (4) In den einzelnen Gebieten sind im ersten Studienabschnitt gemäß Abs. 3, Nr. 1 studienbegleitende Modulprüfungen über den Inhalt der folgenden Module mit dem angegebenen Leistungspunkten abzulegen:
1. Softwaretechnik:
    - 1.1 Programmierung (8 LP)
    - 1.2 Programmiersprachen (4 LP)
    - 1.3 Software Engineering (5 LP)
    - 1.4 Softwaretechnikpraktikum (8 LP)
    - 1.5 Datenbanksysteme (5 LP)
    - 1.6 Gestaltung von Nutzungsschnittstellen (6 LP)
  2. Algorithmen und Komplexität:
    - 2.1 Modellierung (8 LP)
    - 2.2 Datenstrukturen und Algorithmen (9 LP)
    - 2.3 Berechenbarkeit und Komplexität (6 LP)
  3. Computersysteme:
    - 3.1 Digitaltechnik (5 LP)
    - 3.2 Rechnerarchitektur (5 LP)
    - 3.3 Systemsoftware und systemnahe Programmierung (9 LP)
    - 3.4 IT-Sicherheit (5 LP)
  4. Daten und Wissen:  
Für dieses Gebiet sind im ersten Studienabschnitt keine Module vorgesehen.

5. Mathematik:
  - 5.1 Analysis für Informatiker (8 LP)
  - 5.2 Lineare Algebra für Informatiker (8 LP)
  - 5.3 Stochastik für Informatiker (6 LP)

Wurde das Nebenfach Mathematik gewählt, gilt eine abweichende Regelung. Näheres ist der Nebenfachvereinbarung zum Fach Mathematik zu entnehmen, die im Anhang 2 enthalten ist.

- (5) Im zweiten Studienabschnitt sind gemäß Abs. 3 Nr. 2 studienbegleitende Modulprüfungen wie folgt abzulegen:
  1. In jedem der fünf Informatikgebiete (Softwaretechnik, Algorithmen und Komplexität, Computersysteme, Daten und Wissen sowie Vertiefung) ein zugehöriges Wahlpflichtmodul im Umfang von je 6 LP.
  2. Das Modul Schlüsselqualifikation (5 LP) bestehend aus einem Proseminar und Mentoring.
 Außerdem muss das Modul Bachelor-Abschlussarbeit absolviert werden.

- (6) Insgesamt sind im Bachelorstudiengang darüber hinaus
  1. studienbegleitende Modulprüfungen zu Modulen im Nebenfach im Umfang von 18 bis 22 LP sowie
  2. eine studienbegleitende Prüfung im Rahmen des Moduls Studium Generale (abhängig vom gewählten Nebenfach maximal 7 Leistungspunkte) sowie
  3. Nachweise über die qualifizierte Teilnahme an den Veranstaltungen des Studium Generale, in denen keine Prüfungsleistung erbracht wird, zu erbringen. Insgesamt müssen in Nebenfach und Studium Generale zusammen 25 LP absolviert werden. Näheres ist für Standardnebenfächer den Nebenfachvereinbarungen im Anhang 2 dieser Ordnung zu entnehmen. Im Studium Generale dürfen keine Informatikveranstaltungen gewählt werden. Über Ausnahmen entscheidet der Prüfungsausschuss.

## § 11

### Zulassung zur Bachelorprüfung

- (1) Zu Prüfungen in dem Bachelorstudiengang Informatik kann nur zugelassen werden, wer an der Universität Paderborn für den Bachelorstudiengang Informatik eingeschrieben oder gemäß § 52 HG als Zweithörerin bzw. Zweithörer zugelassen ist. Auch während der Prüfungen müssen diese Erfordernisse gegeben sein.
- (2) Zusätzlich zu den in Absatz 1 genannten Voraussetzungen und möglichen modulspezifischen Regelungen gemäß Anhang 1 kann zu den Modulprüfungen des zweiten Studienabschnitts im Hauptfach erst zugelassen werden, wer die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker bestanden hat.

Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.

- (3) Es wird nachdrücklich empfohlen, vor der letzten Prüfung im Hauptfach eine berufspraktische Tätigkeit von mindestens acht Wochen Dauer in einer einschlägigen Umgebung nachzuweisen oder ein Auslandssemester zu absolvieren. Die Hochschule unterstützt die Suche nach einem Studienplatz im Ausland und nach einer Stelle für eine berufspraktische Tätigkeit im Rahmen ihrer Möglichkeiten. Ein Anspruch auf Zuweisung besteht nicht. Die berufspraktische Tätigkeit soll in der Regel erst begonnen werden, wenn die Summe der bestandenen Module des ersten Studienabschnitts 79 LP erreicht hat und die Prüfungen des Gebietes Softwaretechnik (1.4 in § 10 Abs. 4 Nr. 1) bestanden wurden. Dasselbe gilt auch für das Auslandssemester.
- (4) Die Festlegung des Nebenfachs erfolgt 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 bzw. der Kandidat meldet ihre bzw. seine Teilnahme an einer Prüfung im Nebenfach an. Auf Antrag beim Prüfungsausschuss ist ein einmaliger Wechsel des Nebenfachs möglich, auch nach endgültigem Nichtbestehen einer Prüfung im Nebenfach gemäß § 10 Abs. 3 Nr. 3.
- (5) Zum Modul Bachelor-Abschlussarbeit kann nur zugelassen werden, wer alle Module des ersten Studienabschnitts im Hauptfach gemäß § 10 Abs. 3 Nr. 1 und das Modul Schlüsselqualifikation erfolgreich abgeschlossen hat. Die Ausgabe des Themas 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.
- (6) Die Meldung zur Bachelorarbeit ist schriftlich über das Zentrale Prüfungssekretariat an die Vorsitzende bzw. den Vorsitzenden des Prüfungsausschusses zu stellen.
- (7) Die Zulassung zu einer Prüfung ist abzulehnen, wenn die in Abs. 1, 2 und 5 genannten Voraussetzungen nicht erfüllt sind.
- (8) Weitere Voraussetzungen für die Teilnahme an Prüfungen können in den Modulbeschreibungen geregelt werden.

## § 12

### Anmeldung und Prüfungsfristen

- (1) Zu jedem Modul ist eine gesonderte Anmeldung über das Campus Management System der Universität Paderborn erforderlich. Zudem ist zu jeder Prüfung eine gesonderte Anmeldung 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 Prüfungen in Klausurform erfolgt innerhalb der festgelegten Fristen über das Campus Management System der Universität Paderborn. Dasselbe gilt für mündliche Prüfungen, für die vom Veranstalter ein Prüfungsblock von bis zu drei Wochen Dauer festgelegt wird, währenddessen die einzelnen Prüfungen stattfinden (Blockprüfungen), sowie für mündliche Prüfungen ohne Festlegung eines Prüfungsblocks (Individualprüfungen). Der konkrete Prüfungstermin wird dabei von der oder dem Prüfenden vergeben. Mündliche Ersatzprüfungen und Individualprüfungen, bei denen die Veranstaltung nicht im aktuellen Semester angeboten wird, müssen bis spätestens drei Wochen vor dem Prüfungstermin angemeldet werden. Die Anmeldetermine für das Proseminar werden vor der Prüfungsanmeldephase von der bzw. dem Lehrenden festgelegt.

- (3) Die Prüfungen können abgelegt werden, sobald die für die Zulassung erforderlichen Voraussetzungen 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 Teilnahme 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 Bachelorprüfung 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 Lehrveranstaltungen 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 Klausurarbeiten, mündlichen Prüfungen, Projektarbeiten 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. Klausuren:

- In den Klausuren soll die Kandidatin bzw. der Kandidat nachweisen, dass sie bzw. er in einer vorgegebenen Zeit mit den von der bzw. dem Prüfenden zugelassenen Hilfsmitteln Probleme des Faches erkennen und mit geläufigen Methoden lösen kann.
- Jede Klausur muss von mindestens einer bzw. einem Prüfenden gemäß § 9 bewertet werden. § 20 Abs. 5 bleibt unberührt.
- Klausuren dauern in der Regel mindestens 90 und höchstens 180 Minuten.

#### 2. 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 breites Grundlagenwissen verfügt.
- Mündliche Prüfungen werden vor zwei Prüfenden oder vor einer bzw. einem Prüfenden in Gegenwart einer bzw. eines sachkundigen Beisitzenden gemäß § 9 als Einzelprüfungen oder als Gruppenprüfungen von in der Regel nicht mehr als vier Kandidatinnen bzw. Kandidaten abgelegt. § 20 Abs. 5 bleibt unberührt. Vor der Festsetzung der Note beraten die Prüfenden bzw. hört die bzw. der Prüfende die Beisitzende oder den Beisitzenden in Abwesenheit der Kandidatin oder des Kandidaten.
- Mündliche Prüfungen dauern in der Regel mindestens 25 und höchstens 50 Minuten. Bei Gruppenprüfungen verlängert sich die Gesamtprüfungsdauer entsprechend der Kandidatenzahl.
- 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 bekannt zu geben.
- Studierende, die sich in einem späteren Prüfungstermin der gleichen 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.

#### 3. Prüfungsleistung im Softwaretechnikpraktikum:

Im Softwaretechnikpraktikum 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.

#### 4. Prüfungsleistung im Proseminar:

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

- (2) Als Studienleistung können Übungsaufgaben verlangt werden, die in der Regel wöchentlich als Hausaufgaben und/oder Präsenzaufgaben gestellt werden.
- (3) Die qualifizierte Teilnahme wird in der Regel durch eine Praktikumsarbeit mit anschließendem Gespräch nachgewiesen. Eine qualifizierte Teilnahme liegt vor, wenn die erbrachten Leistungen erkennen lassen, dass eine mehr als nur oberflächliche Beschäftigung mit den Gegenständen, die einer Aufgabenstellung zugrunde lagen, stattgefunden hat.
- (4) Im Studium Generale 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).

## § 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 mehr 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 3 und 4 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 Bachelor-Abschlussarbeit (15 LP) besteht aus der Arbeitsplanung (3 LP) und der Bachelorarbeit einschließlich eines Vortrags (12 LP).
- (2) Die Bachelorarbeit 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. Die Aufgabenstellung soll so gestaltet werden, dass sie einem Arbeitsaufwand von neun Wochen Vollzeitarbeit entspricht. Die Arbeit wird studienbegleitend erstellt und muss fünf Monate nach der Ausgabe abgegeben werden. Sie soll einen Umfang von in der Regel nicht mehr als 60 DIN A4-Seiten haben.
- (3) Die Bachelorarbeit wird von einer von der bzw. dem Vorsitzenden des Prüfungsausschusses bestellten Person mit Prüferqualifikation nach § 9 gestellt und betreut. Auf Antrag der Kandidatin bzw. des Kandidaten kann der Prüfungsausschuss auch Prüfungsberechtigte zur Betreuung der Bachelorarbeit 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 Bachelorarbeit 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 Bachelorarbeit erhält. Der Zeitpunkt der Vergabe ist beim Zentralen Prüfungssekretariat aktenkundig zu machen.
- (5) Die Bachelorarbeit 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. Unter der Voraussetzung gleicher Abgabefristen ist dabei auch die Erstellung einer Gruppenarbeit mit Studierenden des Bachelorteilzeitstudiengangs zulässig.
- (6) Das Thema und die Aufgabenstellung der Bachelorarbeit 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 Bachelorarbeit nach der in Satz 2 genannten Frist zurückgegeben, gilt die Bachelorarbeit als nicht bestanden. Die Bearbeitungszeit beginnt mit der Vergabe des neuen Themas nach der 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 zwei Wochen verlängern, wenn die Gründe hierfür mit dem Thema der Arbeit zusammenhängen und die bzw. der Betreuende nach Abs. 3 dies befürwortet.
- (7) 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.
- (8) Die Bachelorarbeit darf nicht, auch nicht auszugsweise, für eine andere abgeschlossene Prüfung angefertigt worden sein.
- (9) Bei der Abgabe der Bachelorarbeit 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.



- (10) Bei Erkrankung innerhalb der Bearbeitungszeit kann auf Antrag der Kandidatin bzw. des Kandidaten die Frist für die Abgabe der Bachelorarbeit 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.

## § 18

### Annahme und Bewertung der Bachelorarbeit

- (1) Die Bachelorarbeit ist fristgemäß beim Zentralen Prüfungssekretariat in zweifacher Ausfertigung (maschinenschriftlich, paginiert und gebunden) sowie zusätzlich einmal in elektronischer Form durch ein physisches Medium abzugeben; der Abgabezeitpunkt ist beim Zentralen Prüfungssekretariat aktenkundig zu machen. Bei Zustellung der Arbeit mit der Post ist der Zeitpunkt der Einlieferung bei der Post (Poststempel) maßgebend. Wird die Arbeit nicht fristgemäß abgegeben, gilt sie als mit „mangelhaft“ (5,0) bewertet.
- (2) Die Bachelorarbeit einschließlich des Vortrags ist von zwei Prüfenden gemäß § 9 zu begutachten und zu bewerten. Der Vortrag von ca. 30 Minuten Dauer findet in der Regel spätestens vier Wochen nach dem Abgabezeitpunkt 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 Bachelorarbeit (ohne Vortrag) 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 Bachelorarbeit ist gleichzeitig die Note des Moduls Bachelor-Abschlussarbeit.
- (4) Die Bewertung der Bachelorarbeit ist den Studierenden spätestens sechs Wochen nach Abgabe der Bachelorarbeit im Campus Management System der Universität Paderborn bekannt zu geben.

## § 19

### Bewertung der Bachelorprüfung und Bildung der Gesamtnote

- (1) Die Bachelorprüfung ist bestanden, wenn alle Modulprüfungen und die Bachelorarbeit mit mindestens „ausreichend“ (4,0) bewertet sind.

- (2) Die Gesamtnote wird gebildet, indem alle Modulnoten gewichtet werden. Sie errechnet sich aus dem gewichteten Durchschnitt der Noten aus den Modulprüfungen des Hauptfachs einschließlich Mathematik, sowie den Noten der Modulprüfungen des Nebenfachs und des Studium Generale. Für die Gewichtung werden – unbenommen der tatsächlichen LP
- die LP des ersten Studienabschnitts einfach, wobei das Modul 1.4 Softwaretechnikpraktikum mit 4 Gewichtspunkten gerechnet wird (101 Gewichtspunkte),
  - die des zweiten Abschnitts mit Ausnahme der Abschlussarbeit doppelt (70 Gewichtspunkte),
  - die des Moduls Bachelor-Abschlussarbeit vierfach (48 Gewichtspunkte)
  - die des jeweiligen Nebenfachs mit 30 Gewichtspunkten
  - und die des Studium Generale mit 7 Gewichtspunkten gezählt.

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 nichtbestandene Prüfung kann zweimal wiederholt werden. Die erste oder zweite Wiederholung einer Prüfung in Klausurform wird auf Wunsch der Kandidatin oder des Kandidaten als mündliche Ersatzprüfung abgehalten. Für die Abnahme und Bewertung der Ersatzprüfung gelten die §§ 14 und 15 entsprechend. Die Ersatzprüfung kann nur mit den Noten „ausreichend“ (4,0) oder „mangelhaft“ (5,0) bewertet 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 endgültig 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 Bachelor-Abschlussarbeit kann bei der Bewertung „mangelhaft“ einmal wiederholt werden. Dabei ist ein neues Thema zu stellen. Eine zweite Wiederholung ist ausgeschlossen. Eine Rückgabe des Themas der Bachelorarbeit 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 Bachelorarbeit 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. Sie werden bei der Notenbildung der Bachelorprüfung 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 Prüfungen in Klausurform oder von Individualprüfungen und mündlichen Ersatzprüfungen 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 das Softwaretechnikpraktikum und das Proseminar werden vor der Prüfungsanmeldephase von der bzw. dem Lehrenden 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. 1 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 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 Elternzeitgesetzes auslösen würden und legt unter Berücksichtigung des Einzelfalls die Termine und Fristen fest. Die Abgabefrist der Bachelorarbeit 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 Bachelorprüfung bestanden ist und alle Module erfolgreich abgeschlossen sind. Die Bachelorprüfung ist bestanden, wenn alle Modulprüfungen nach § 10 mindestens mit der Note „ausreichend“ (4,0) benotet wurden.
- (2) Die Bachelorprüfung ist endgültig nicht bestanden, wenn ein Modul endgültig nicht bestanden ist oder die Bachelorarbeit zum zweiten Mal mit der Note „mangelhaft“ (5,0) bewertet worden ist.
- (3) Der Bescheid über eine endgültig nicht bestandene Bachelorprüfung 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 eine Kandidatin bzw. ein Kandidat die Bachelorprüfung 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 Bachelorprüfung endgültig nicht bestanden ist.
- (5) Studierenden, welche 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, Angaben zum Nebenfach, 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 Fachstudierendauer aufgeführt sind. Das Transcript of Records enthält Angaben über die Leistungspunkte und die erzielten Noten zu den absolvierten Modulen und zu der Bachelorarbeit. Es enthält des Weiteren das Thema der Bachelorarbeit und die erzielte Gesamtnote der Bachelorprüfung.
- (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.

**§ 25****Bachelorurkunde**

- (1) Gleichzeitig mit dem Zeugnis über den bestandenen Bachelorabschluss wird der Kandidatin bzw. dem Kandidaten eine Urkunde mit dem Datum des Zeugnisses ausgehändigt. Darin wird die Verleihung des Bachelorgrades gemäß § 2 beurkundet.
- (2) Die Bachelorurkunde 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 Bachelorurkunde 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 Lehrende 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 Bachelorarbeit, 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 Bachelorprüfung**

- (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 bzw. dem Betroffenen Gelegenheit zur Äußerung zu geben.
- (4) Das unrichtige Zeugnis ist einzuziehen und gegebenenfalls ein neues zu erteilen. Eine Entscheidung nach Abs. 1 und Abs. 2 Satz 2 ist nach einer Frist von fünf Jahren nach Ausstellung des Zeugnisses ausgeschlossen.
- (5) Ist die Prüfung insgesamt für nicht bestanden erklärt worden, sind der Bachelorgrad abzuerkennen und die entsprechende Bachelorurkunde einzuziehen. Die Aberkennung des Bachelorgrades ist nur innerhalb von fünf Jahren seit dem Zeitpunkt der Gradverleihung zulässig.

### **§ 28**

#### **Aberkennung des Bachelorgrades**

Der Bachelorgrad 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 Bachelorstudiengang 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 Bachelorstudiengang Informatik eingeschrieben worden sind, können ihre Bachelorprüfung einschließlich Wiederholungsprüfungen letztmalig im Wintersemester 2021/2022 nach der Prüfungsordnung in der Fassung vom 28. Februar 2013 (AM.Uni.Pb. 09.13), zuletzt geändert durch Satzung vom 15. Januar 2016 (AM.Uni.Pb. 02.16), ablegen. Engere Fristen aus älteren Übergangsbestimmungen bleiben unberührt. Ab dem Sommersemester 2022 wird die Bachelorprüfung 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 Bachelorprüfungsordnung vom 28. Februar 2013 (AM.Uni.Pb. 09.13), zuletzt geändert durch Satzung vom 15. Januar 2016 (AM.Uni.Pb. 02.16), 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



## Anhänge

### Anhang 1: Module, Zulassungsvoraussetzungen und Prüfungsformen und -modalitäten

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 und zu Teilnahmevoraussetzungen bleiben hiervon unberührt.

Modul Lehrveranstaltung (LV)	LP Modul SWS LV	Prüfungsformen	Bemerkung
<b>Programmierung</b>	<b>8</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Softwaretechnik Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Programmierung	4+2		
<b>Programmiersprachen</b>	<b>4</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Softwaretechnik Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Programmiersprachen	2+1		
<b>Software Engineering</b>	<b>5</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Softwaretechnik Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung Qualifizierte Teilnahme am Praktikum
Softwareentwurf	2+1		
Praktikum: Software Engineering	1		
<b>Softwaretechnikpraktikum</b>	<b>8</b>	1. Phasenbezogene Modulteilprüfung über in der Regel 5 Projekte (Abgabe von Software und Dokumentation) und 2. Klausur als weitere Modulteilprüfung. Dabei gehen die Bewertungen der phasenbezogenen Modulteilprüfung mit insgesamt 70% und die Klausur mit 30% in die Note ein.	Pflichtmodul im Gebiet Softwaretechnik
<b>Datenbanksysteme</b>	<b>5</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Softwaretechnik Voraussetzung zur Teilnahme an der Modulprüfung:
Datenbanksysteme	3+1		

<b>Modul</b> Lehrveranstaltung (LV)	<b>LP Modul</b> SWS LV	<b>Prüfungsformen</b>	<b>Bemerkung</b>
			Studienleistung
<b><i>Gestaltung von Nutzungsschnittstellen</i></b>	<b>6</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Softwaretechnik Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Gestaltung von Nutzungsschnittstellen	4+1		
<b><i>Modellierung</i></b>	<b>8</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Algorithmen und Komplexität Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Modellierung	4+2		
<b><i>Datenstrukturen und Algorithmen</i></b>	<b>9</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Algorithmen und Komplexität Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung Qualifizierte Teilnahme am Praktikum
Datenstrukturen und Algorithmen	4+2		
Praktikum: Datenstrukturen und Algorithmen	1		
<b><i>Berechenbarkeit und Komplexität</i></b>	<b>6</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Algorithmen und Komplexität Teilnahmevoraussetzung: erfolgreicher Abschluss der Module Modellierung und Datenstrukturen und Algorithmen Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Berechenbarkeit und Komplexität	3+2		
<b><i>Digitaltechnik</i></b>	<b>5</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Computersysteme
Digitaltechnik	2+2		
<b><i>Rechnerarchitektur</i></b>	<b>5</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Computersysteme
Rechnerarchitektur	2+2		
<b><i>Systemsoftware und systemnahe Programmierung</i></b>	<b>9</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Computersysteme Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung Qualifizierte Teilnahme am Praktikum
Systemsoftware und systemnahe Programmierung	4+2		
Praktikum: Systemsoftware und systemnahe Programmierung	1		
<b><i>IT-Sicherheit</i></b>	<b>5</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Computersysteme Teilnahmevoraussetzung:
IT-Sicherheit	2+2		

Modul Lehrveranstaltung (LV)	LP Modul SWS LV	Prüfungsformen	Bemerkung
			erfolgreicher Abschluss des Moduls Systemsoftware und systemnahe Programmierung Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
<b><i>Analysis für Informatiker</i></b>	<b>8</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Mathematik Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Analysis für Informatiker	4+2		
<b><i>Lineare Algebra für Informatiker</i></b>	<b>8</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Mathematik Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Lineare Algebra für Informatiker	4+2		
<b><i>Stochastik für Informatiker</i></b>	<b>6</b>	Klausur als Modulabschlussprüfung	Pflichtmodul im Gebiet Mathematik Teilnahmevoraussetzung: erfolgreicher Abschluss der Module Analysis für Informatiker und Lineare Algebra für Informatiker Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Stochastik für Informatiker	3+2		
<b><i>Informatikgebiet Softwaretechnik</i></b>	<b>6</b>	Klausur oder mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul im 2. Studienabschnitt Teilnahmevoraussetzung: vgl. § 11 Abs. 2 Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Interaktionsgestaltung</li> <li>• Logik und Deduktion</li> <li>• Modellbasierte Softwareentwicklung</li> <li>• Programmiersprachen und Übersetzer</li> <li>• Secure Software Engineering</li> <li>• Softwaremodellierung mit formalen Methoden</li> </ul>	3+2		
<b><i>Informatikgebiet Algorithmen und Komplexität</i></b>	<b>6</b>	Klausur oder mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul im 2. Studienabschnitt Teilnahmevoraussetzung: vgl. § 11 Abs. 2 Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Einführung in Kryptographie</li> <li>• Grundlegende Algorithmen</li> <li>• Komplexitätstheorie</li> <li>• Parallelität und Kommunikation</li> <li>• Verteilte Algorithmen und Datenstrukturen</li> </ul>	3+2		

Modul Lehrveranstaltung (LV)	LP Modul SWS LV	Prüfungsformen	Bemerkung
<b><i>Informatikgebiet Computersysteme</i></b>	6	Klausur oder mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul im 2. Studienabschnitt Teilnahmevoraussetzung: vgl. § 11 Abs. 2 Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Betriebssysteme</li> <li>• Eingebettete Systeme</li> <li>• Rechnernetze</li> <li>• Verteilte Systeme</li> </ul>	3+2		
<b><i>Informatikgebiet Daten und Wissen</i></b>	6	Klausur oder mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul im 2. Studienabschnitt Teilnahmevoraussetzung: vgl. § 11 Abs. 2 Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Computer Graphics Rendering</li> <li>• Data Mining</li> <li>• Databases and Information Systems</li> <li>• Grundlagen wissensbasierter Systeme</li> </ul>	3+2		
<b><i>Informatikgebiet Vertiefung</i></b>	6	Klausur oder mündliche Prüfung als Modulabschlussprüfung	Wahlpflichtmodul im 2. Studienabschnitt Teilnahmevoraussetzung: vgl. § 11 Abs. 2 Voraussetzung zur Teilnahme an der Modulprüfung: Studienleistung
Eines der Module <ul style="list-style-type: none"> <li>• Betriebssysteme</li> <li>• Computer Graphics Rendering</li> <li>• Data Mining</li> <li>• Databases and Information Systems</li> <li>• Einführung in Kryptographie</li> <li>• Eingebettete Systeme</li> <li>• Grundlagen wissensbasierter Systeme</li> <li>• Grundlegende Algorithmen</li> <li>• Interaktionsgestaltung</li> <li>• Komplexitätstheorie</li> <li>• Logik und Deduktion</li> <li>• Modellbasierte Softwareentwicklung</li> <li>• Parallelität und Kommunikation</li> <li>• Programmiersprachen und Übersetzer</li> <li>• Rechnernetze</li> <li>• Secure Software Engineering</li> <li>• Softwaremodellierung mit formalen Methoden</li> <li>• Verteilte Algorithmen und Datenstrukturen</li> <li>• Verteilte Systeme</li> </ul>	3+2		
<b><i>Schlüsselqualifikation</i></b>	5	Seminarvortrag und schriftliche Ausarbeitung, Note ergibt die Modulnote	Pflichtmodul im 2. Studienabschnitt Qualifizierte Teilnahme: Regelmäßige Einreichung (einmal pro Semester) des Studienplanungsbogens beim
Proseminar	4		
Mentoring	1		

Modul Lehrveranstaltung (LV)	LP Modul SWS LV	Prüfungsformen	Bemerkung
			Mentor und Abgabe des ausgefüllten Studiumbewertungsbogens
<i>Nebenfach und Studium Generale</i>	25		Nebenfach und Studium Generale ergänzen sich auf insgesamt 25 LP
Nebenfach	18 - 22	Modulabschlussprüfung gemäß der Nebenfachvereinbarung	Wählbares Nebenfach aus dem Katalog der Standardnebenfächer bzw. Nicht-Standardnebenfach auf individuellen Antrag
Studium Generale	3-7	Prüfung im Studium Generale	Qualifizierte Teilnahme bei übrigen LV
<i>Bachelor-Abschlussarbeit</i>	15	Prüfung gemäß §17 und §18	Besteht aus Arbeitsplanung (3 LP), Anfertigung der Bachelorarbeit einschließlich Abschlussvortrag (12 LP)

## Anhang 2: Nebenfachvereinbarungen für die Standardnebenfächer im Bachelorstudiengang 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

3. Semester	Grundlagen der Elektrotechnik A (GET A)	MP	6
4.-6. Semester	Signaltheorie	MP	6
	Systemtheorie	MP	6
		<b>SUMME</b>	<b>18</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale 7 LP absolviert werden.

Hinweis: Statt GET A kann auch das Modul „Grundlagen der Elektrotechnik für Maschinenbau“ absolviert werden.

### 2. Mathematik

Das Nebenfach Mathematik wird anders als die anderen Standardnebenfächer vom ersten Semester an studiert.

1. Semester	Lineare Algebra 1 (*)	MP	9
2. Semester	Lineare Algebra 2	MP	9
3. Semester	Analysis 1 (*)	MP	9
4. bis 6. Semester	Analysis 2	MP	9
	oder Numerik 1	MP	9
	oder Algebra	MP	9
		<b>Summe</b>	<b>36</b>

(\*) Dafür entfallen die Module 1.5.1 Analysis für Informatiker und 1.5.2 Lineare Algebra für Informatiker (je 8 LP)

Durch den Abzug der Module 1.5.1 und 1.5.2 hat dieses Nebenfach einen Umfang von 20 LP. Dies muss mit dem Studium Generale auf 25 LP aufgefüllt werden, d.h. es müssen 5 LP im Studium Generale absolviert werden. Das Modul 1.5.3 Stochastik für Informatiker (6 LP) muss absolviert werden, kann aber optional durch die Veranstaltung „Fundamente der Stochastik 1“ (5 LP) ersetzt werden. Bei dieser Variante muss 1 weiterer LP im Studium Generale absolviert werden.

### 3. Medienwissenschaft

3.-6. Semester	<b>Basismodul Medienanalyse</b> (Musik, visuelle Medien, Text in den Medien, digitale Medien)		
	Einführung	SP	4
	Seminar/Lehrveranstaltung	SP	4
	Seminar/Lehrveranstaltung	MP	4
	<b>Reduziertes Modul Mediensoziologie/-pädagogik/-psychologie</b>		
	Einführung	SP	4
	Seminar/Lehrveranstaltung	MP	6
		<b>SUMME</b>	<b>22</b>

Bei Wahl dieses Nebenfachs muss das Studium Generale im Umfang von 3 LP absolviert werden.

### 4. Philosophie

3. Semester	Basismodul 1: Grundlagen und Methoden der Philosophie	Überblicksveranstaltung Seminar	qT qT	6
4.–6. Semester	Basismodul 2: Praktische Philosophie	Überblicksveranstaltung Seminar	qT qT	8
		Modulprüfung	MP	
	Basismodul 3: Theoretische Philosophie	Überblicksveranstaltung Seminar	qT qT	8
		Modulprüfung	MP	
			<b>SUMME</b>	<b>22</b>

Bei Wahl dieses Nebenfachs muss das Studium Generale im Umfang von 3 LP absolviert werden.

### 5. Psychologie

3. und 4. Semester	Basismodul Kognitionspsychologie und psychologische Methoden		
	Einführung in die Psychologie mit Tutorium	qT	4
	Veranstaltung zur Entwicklungspsychologie oder Einführung in die Kognitionspsychologie	qT	2
	Empirische Methoden (Seminar)	MP	4
5. und 6. Semester	Aufbaumodul Arbeits- und Organisationspsychologie		
	Arbeits- und Personalpsychologie oder Organisationspsychologie	qT	2
	Arbeits- und Organisationspsychologie	qT	4

	Portfolioprüfung		2
	oder		
5. und 6. Semester	Aufbaumodul Kognitionspsychologie		
	Kognitionspsychologisches Seminar	qT	2
	Experimentalpsychologisches oder empirisches Praktikum	qT	4
	Portfolioprüfung	MP	2
		<b>SUMME</b>	<b>18</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale 7 LP absolviert werden.

## 6. Wirtschaftsinformatik

3. Semester	Veranstaltungen aus dem Bachelor-Vertiefungsbereich Wirtschaftsinformatik (*)	MP	10
4. Semester			
5. Semester			
6. Semester	Veranstaltungen aus dem Bachelor-Vertiefungsbereich Wirtschaftsinformatik (*)	MP	10
		<b>SUMME</b>	<b>20</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale 5 LP absolviert werden.

(\*) Wählbar sind alle Veranstaltungen für Studierende im Bachelorstudiengang Wirtschaftsinformatik des 3. bis 6. Semesters

## 7. Wirtschaftswissenschaften

3. Semester	Modul M.184.1101: Grundzüge der BWL A	MP	10
4. Semester			
5. Semester			
6. Semester	M.184.1201: Grundzüge der BWL B und des Privatrechts	MP	10
		<b>SUMME</b>	<b>20</b>

Bei Wahl dieses Nebenfachs müssen im Studium Generale 5 LP absolviert werden.



## Anhang 3: Studienverlaufsplan

## Studienplan Bachelor Informatik

Sem.	Erster Studienabschnitt				
1.	GP (8/4+2) PS (4/2+1)	Mod (8/4+2)		Ana (8/4+2)	
2.	SE (5/2+1+1) DBS (5/2+2)	DuA (9/4+2+1)	DT (5/2+2)	LA (8/4+2)	
3.	SWTPRA (8)	BuK (6/3+2)	RA (5/2+2)	St (6/3+2)	
4.	GNS (6/4+1)	AuK (6/3+2)	SSP (9/4+2+1)		NF + SG (z.B. 14)
Nach 4.	Auslandssemester				
Sem.	Zweiter Studienabschnitt (exemplarische Anordnung)				
5.u.6.	SWT (6/3+2) Vertiefung (6/3+2)	ITS (5/2+2) CoSy (6/3+2)	DuW (6/3+2)	Proseminar (4) Mentoring (1)	NF + SG (z.B. 11)
	Bachelor-Arbeit: Arbeitsplanung (3) Durchführung (12)		Nebenfach und Studium Generale (25)		

## Abkürzungen Module 1. Studienabschnitt

Name	Abkürzung
Modellierung	Mod
Programmierung	GP
Programmiersprachen	PS
Analysis für Informatiker	Ana
Software Engineering	SE
Datenbanksysteme	DBS
Datenstrukturen und Algorithmen	DuA
Digitaltechnik	DT
Lineare Algebra für Informatiker	LA
Softwaretechnikpraktikum	SWTPRA
Berechenbarkeit und Komplexität	BuK
Rechnerarchitektur	RA
Stochastik für Informatiker	St
Gestaltung von Nutzungsschnittstellen	GNS
Systemsoftware und systemnahe Programmierung	SSP
IT-Sicherheit	ITS
Nebenfach	NF
Studium Generale	SG

### Abkürzungen Module 2. Studienabschnitt:

- SWT:= Softwaretechnik (bisher SWT & IS, IS Anteil jetzt in DuW)
- AuK:= Algorithmen und Komplexität (bisher MuA)
- CoSy:= Computersysteme (bisher ESS)
- DuW:= Daten und Wissen (neu)

# Inhaltsverzeichnis

<b>1</b>	<b>Beschreibung des Studiengangs Bachelor 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	Erster Studienabschnitt . . . . .	11
2.2	Informatikgebiet Algorithmen und Komplexität . . . . .	13
2.3	Informatikgebiet Computer Systeme . . . . .	14
2.4	Informatikgebiet Daten und Wissen . . . . .	15
2.5	Informatikgebiet Softwaretechnik . . . . .	16
2.6	Informatikgebiet Vertiefung . . . . .	17
<b>3</b>	<b>Module</b>	<b>19</b>
3.1	Pflichtmodul: Analysis für Informatiker . . . . .	20
3.2	Wahlpflichtmodul: Angriffssicherer Softwareentwurf . . . . .	23
3.3	Pflichtmodul: Bachelor-Abschlussarbeit . . . . .	25
3.4	Pflichtmodul: Berechenbarkeit und Komplexität . . . . .	28
3.5	Wahlpflichtmodul: Betriebssysteme . . . . .	30
3.6	Wahlpflichtmodul: Computer Graphics Rendering . . . . .	32
3.7	Wahlpflichtmodul: Data Mining . . . . .	35
3.8	Wahlpflichtmodul: Databases and Information Systems . . . . .	37
3.9	Pflichtmodul: Datenbanksysteme . . . . .	40
3.10	Pflichtmodul: Datenstrukturen und Algorithmen . . . . .	43
3.11	Pflichtmodul: Digitaltechnik . . . . .	46
3.12	Wahlpflichtmodul: Einführung in Kryptographie . . . . .	48
3.13	Wahlpflichtmodul: Eingebettete Systeme . . . . .	50
3.14	Pflichtmodul: Gestaltung von Nutzungsschnittstellen . . . . .	53
3.15	Wahlpflichtmodul: Grundlagen Wissensbasierter Systeme . . . . .	55
3.16	Wahlpflichtmodul: Grundlegende Algorithmen . . . . .	57
3.17	Wahlpflichtmodul: Interaktionsgestaltung . . . . .	60
3.18	Pflichtmodul: IT Sicherheit . . . . .	63
3.19	Wahlpflichtmodul: Komplexitätstheorie . . . . .	65
3.20	Pflichtmodul: Lineare Algebra für Informatiker . . . . .	67
3.21	Wahlpflichtmodul: Logik und Deduktion . . . . .	70
3.22	Wahlpflichtmodul: Modellbasierte Softwareentwicklung . . . . .	72

3.23	Pflichtmodul: Modellierung	74
3.24	Wahlpflichtmodul: Parallelität und Kommunikation	76
3.25	Pflichtmodul: Programmiersprachen	78
3.26	Wahlpflichtmodul: Programmiersprachen und Übersetzer	80
3.27	Pflichtmodul: Programmierung	83
3.28	Pflichtmodul: Rechnerarchitektur	86
3.29	Wahlpflichtmodul: Rechnernetze	88
3.30	Pflichtmodul: Schlüsselqualifikation	90
3.31	Pflichtmodul: Software Engineering	93
3.32	Wahlpflichtmodul: Softwaremodellierung mit Formalen Methoden	96
3.33	Pflichtmodul: Softwaretechnikpraktikum	99
3.34	Pflichtmodul: Stochastik für Informatiker	102
3.35	Pflichtmodul: Studium Generale	105
3.36	Pflichtmodul: Systemsoftware und systemnahe Programmierung	107
3.37	Wahlpflichtmodul: Verteilte Algorithmen und Datenstrukturen	110
3.38	Wahlpflichtmodul: Verteilte Systeme	112
<b>A</b>	<b>Überblickstabellen</b>	<b>115</b>
A.1	Studienrichtungen und Module	116
A.2	Module und Lehrveranstaltungen	117

# Kapitel 1

## Beschreibung des Studiengangs Bachelor Informatik

Der Bachelor-Studiengang Informatik ist in zwei Studienabschnitte aufgeteilt. Er verfolgt das Ziel, breites Wissen in der Informatik zu vermitteln. Um den Studierenden bereits im Bachelor-Studiengang die Möglichkeit zu geben, individuellen Interessen zu folgen, bestehen nach einer Grundausbildung mit Kerninhalten im ersten Studienabschnitt, Wahlmöglichkeiten innerhalb der einzelnen Module im zweiten Studienabschnitt.

## 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**

### **Proseminar plus wählbare Veranstaltung**

**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.

**Klausur** In einer Klausuren soll die Kandidatin bzw. der Kandidat nachweisen, dass sie bzw. er in einer vorgegebenen Zeit mit den von der bzw. dem Prüfenden zugelassenen Hilfsmitteln Probleme des Faches erkennen und mit geläufigen Methoden lösen kann. Klausuren dauern in der Regel mindestens 90 und höchstens 180 Minuten.

**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.

**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

### **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 infromatische 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 Erster Studienabschnitt

Studienrichtung	Erster Studienabschnitt
Koordination	Prof. Dr. rer. nat. Johannes Blömer Codes und Kryptographie Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Analysis für Informatiker (S. 20)</li> <li>• Berechenbarkeit und Komplexität (S. 28)</li> <li>• Datenbanksysteme (S. 40)</li> <li>• Datenstrukturen und Algorithmen (S. 43)</li> <li>• Digitaltechnik (S. 46)</li> <li>• Gestaltung von Nutzungsschnittstellen (S. 53)</li> <li>• IT Sicherheit (S. 63)</li> <li>• Lineare Algebra für Informatiker (S. 67)</li> <li>• Modellierung (S. 74)</li> <li>• Programmiersprachen (S. 78)</li> <li>• Programmierung (S. 83)</li> <li>• Rechnerarchitektur (S. 86)</li> <li>• Schlüsselqualifikation (S. 90)</li> <li>• Software Engineering (S. 93)</li> <li>• Softwaretechnikpraktikum (S. 99)</li> <li>• Stochastik für Informatiker (S. 102)</li> <li>• Systemsoftware und systemnahe Programmierung (S. 107)</li> </ul>

Beschreibung

Im ersten Studienabschnitt werden die wesentlichen Grundkenntnisse und Fähigkeiten vermittelt, die jeder Absolvent der Informatik kennen und beherrschen sollte. Der erste Studienabschnitt besteht nur aus Pflichtveranstaltungen, die mit einer Ausnahme in den ersten vier Fachsemestern liegen.

## 2.2 Informatikgebiet Algorithmen und Komplexität

Studienrichtung	Informatikgebiet Algorithmen und Komplexität
Koordination	Prof. Dr. rer. nat. Johannes Blömer Codes und Kryptographie Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Einführung in Kryptographie (S. 48)</li> <li>• Grundlegende Algorithmen (S. 57)</li> <li>• Komplexitätstheorie (S. 65)</li> <li>• Parallelität und Kommunikation (S. 76)</li> <li>• Verteilte Algorithmen und Datenstrukturen (S. 110)</li> </ul>
Beschreibung	<p>Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus für den Rechner verstehbar. Algorithmen spielen daher eine zentrale Rolle in der Informatik. Deshalb steht im Mittelpunkt des Bachelormoduls Algorithmen und Komplexität die Klassifizierung von Problemen bezüglich ihrer algorithmischen Komplexität. Als Maße für Komplexität werden insbesondere Laufzeit und Speicherbedarf, aber auch z.B. Parallelisierbarkeit herangezogen. Module dieses Gebiets behandeln sowohl die Entwicklung und Analyse effizienter Algorithmen und algorithmischer Techniken, als auch die Untersuchung der Problem-inhärenten Komplexität, d.h. den Nachweis unterer Komplexitätsschranken und den Komplexitätsvergleich von Problemen. Weiter ergänzt wird das Gebiet durch ein Modul zur Kryptographie. Hier wird die inhärente Schwierigkeit von Problemen, wie sie die Komplexitätstheorie nachzuweisen versucht, positiv etwa für den Entwurf sicherer Verschlüsselungsverfahren genutzt.</p>

## 2.3 Informatikgebiet Computer Systeme

Studienrichtung	Informatikgebiet Computer Systeme
Koordination	Prof. Dr. Marco Platzner Technische Informatik Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Betriebssysteme (S. 30)</li> <li>• Eingebettete Systeme (S. 50)</li> <li>• Rechnernetze (S. 88)</li> <li>• Verteilte Systeme (S. 112)</li> </ul>
Beschreibung	
<p>Jede Informatikanwendung benötigt ein technisches System, auf der sie ausgeführt wird. Beispiele für solche Systeme sind auf bestimmte Anwendungsklassen zugeschnittene Computersysteme wie eingebettete Systeme oder verteilte Systeme. Wesentliche Komponenten von Computersystemen sind neben der Rechner-Hardware auch die Betriebssysteme eines Rechners oder die technische Infrastruktur für die Vernetzung mehrerer Rechner.</p> <p>Die Eignung und Qualität eines Gesamtsystems hängt wesentlich davon ab, dass die Eigenheiten, Vorteile und Einschränkungen dieser technischen Systeme sinnvoll ausgenutzt werden. Zudem entstehen aus der Evolution von Infrastrukturen neue Anwendungsklassen und tragen damit wesentlich zur Innovation in der Informatik und ihren Anwendungen bei. Wesentliche Beispiele sind hier zweifellos das Internet oder der Mobilfunk samt Smartphones, deren tief greifender Einfluss kaum überschätzt werden kann.</p> <p>Das Verständnis solcher Systeme und die Fähigkeit, solche Systeme gezielt zu benutzen und weiterzuentwickeln, ist daher eine wesentliche Kernkompetenz eines Informatikers. In diesem Gebiet werden diese Fertigkeiten erworben und vertieft. Das Gebiet baut insbesondere auf den Modulen Digitaltechnik, Rechnerarchitektur sowie Systemsoftware und systemnahe Programmierung auf. Die dortigen Grundlagen werden hier methodisch vertieft, die zielgeleitete Konstruktion technischer Systeme wird untersucht und Verfahren zur Bewertung der Leistungsfähigkeit und Eignung solcher Systeme abgeleitet. Dabei fokussieren die einzelnen Module, die innerhalb dieses Gebiets gewählt werden können, auf einzelne Teilbereiche von Computersystemen: Vernetzung von Rechnern, verteilte Systeme, Betriebssysteme und eingebettete Systeme.</p>	

## 2.4 Informatikgebiet Daten und Wissen

Studienrichtung	Informatikgebiet Daten und Wissen
Koordination	Prof. Dr. Stefan Böttcher Electronic Commerce und Datenbanken Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Computer Graphics Rendering (S. 32)</li> <li>• Data Mining (S. 35)</li> <li>• Databases and Information Systems (S. 37)</li> <li>• Grundlagen Wissensbasierter Systeme (S. 55)</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 Gebiet kombiniert die Themen Daten und Wissen im Sinne einer modern ausgerichteten KI und vermittelt methodische Grundlagen intelligenter Systeme. Die Inhalte des Gebiets erstrecken sich von Aspekten der Wissensrepräsentation über die automatisierte Wissensverarbeitung bis zum Erwerb von Wissen aus Daten. Der effektive Umgang mit Daten durch systematische Organisation und Verarbeitung großer Datenbestände wird in Vorlesungen zum Thema Datenbanken und Informationssysteme erlernt. Ein weiteres Thema dieses Gebiets, die Computergraphik und grafische Datenverarbeitung, widmet sich einer speziellen Form von Daten, nämlich digitalen Bildern, abstrakten Beschreibungen großer 3D Szenen, sowie der visuellen Kommunikation zwischen Mensch und Maschine. Die wichtige Echtzeit-Verarbeitung oft extrem umfangreicher grafischer Datenmengen wird durch die Implementierung von Code auf GPUs (Graphics Processing Units) vermittelt.</p>



## 2.5 Informatikgebiet Softwaretechnik

Studienrichtung	Informatikgebiet Softwaretechnik
Koordination	Prof. Dr. Heike Wehrheim Spezifikation und Modellierung von Softwaresystemen Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Angriffssicherer Softwareentwurf (S. 23)</li> <li>• Interaktionsgestaltung (S. 60)</li> <li>• Logik und Deduktion (S. 70)</li> <li>• Modellbasierte Softwareentwicklung (S. 72)</li> <li>• Programmiersprachen und Übersetzer (S. 80)</li> <li>• Softwaremodellierung mit Formalen Methoden (S. 96)</li> </ul>
Beschreibung	<p>Das Gebiet vermittelt einen breiten Überblick über die wichtigsten Konzepte, Notationen und Methoden der Softwaretechnik und ihrer formalen und mathematischen Grundlagen. Die vermittelten Kenntnisse sollen die Studierenden in die Lage versetzen, Softwaresysteme unter vorgegebenen technischen und ökonomischen Randbedingungen zu entwickeln. Darüber hinaus sollen die Studierenden das wissenschaftliche Handwerkzeug beherrschen, um sich im späteren Berufsleben in zukünftige Techniken einzuarbeiten.</p>

## 2.6 Informatikgebiet Vertiefung

Studienrichtung	Informatikgebiet Vertiefung
Koordination	Prof. Dr. Gerd Szwillus Mensch-Computer Interaktion Informatik
Enthaltene Module	<ul style="list-style-type: none"> <li>• Angriffssicherer Softwareentwurf (S. 23)</li> <li>• Betriebssysteme (S. 30)</li> <li>• Computer Graphics Rendering (S. 32)</li> <li>• Data Mining (S. 35)</li> <li>• Databases and Information Systems (S. 37)</li> <li>• Einführung in Kryptographie (S. 48)</li> <li>• Eingebettete Systeme (S. 50)</li> <li>• Grundlagen Wissensbasierter Systeme (S. 55)</li> <li>• Grundlegende Algorithmen (S. 57)</li> <li>• Interaktionsgestaltung (S. 60)</li> <li>• Komplexitätstheorie (S. 65)</li> <li>• Logik und Deduktion (S. 70)</li> <li>• Modellbasierte Softwareentwicklung (S. 72)</li> <li>• Parallelität und Kommunikation (S. 76)</li> <li>• Programmiersprachen und Übersetzer (S. 80)</li> <li>• Rechnernetze (S. 88)</li> <li>• Softwaremodellierung mit Formalen Methoden (S. 96)</li> <li>• Verteilte Algorithmen und Datenstrukturen (S. 110)</li> <li>• Verteilte Systeme (S. 112)</li> </ul>

Beschreibung

Dieses Gebiet erlaubt Studierenden eine Vertiefung in einem der vier Gebiete der Informatik (Algorithmen und Komplexität, Computersysteme, Daten und Wissen, Softwaretechnik).

# Kapitel 3

# Module

### 3.1 Pflichtmodul: Analysis für Informatiker

Modulname	Analysis für Informatiker / Calculus for Computer Science
Workload	240 h
Leistungspunkte	8 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Analysis für Informatiker : 1</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Analysis für Informatiker: Vorlesung ( 60h / 150h / DE / WS / 450 ) Analysis für Informatiker: Übung ( 30h / 0h / DE / WS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Analysis für Informatiker:	
<b>Inhalte</b>	
Analysis für Informatiker: Das Modul besteht aus der Vorlesung Analysis (für Informatiker) Gliederung „Analysis (V4+Ü2)“ Kapitel I Grundbegriffe 1. Mengen und Abbildungen 2. Vollständige Induktion und Rekursion, Kombinatorik 3. Elementare Zahlentheorie 4. Reelle Zahlen, Körper 5. Die komplexen Zahlen Kapitel II Analysis 1. Konvergenz von Folgen 2. Konvergenz von Reihen und Potenzreihen 3. Stetigkeit 4. Exponentialfunktion und trigonometrische Funktionen 5. Polarkoordinaten, Einheitswurzeln und der Fundamentalsatz der Algebra 6. Differenzierbarkeit 7. Lokale Extrema, Taylor-Formel, Taylor-Reihen 8. Integrierbarkeit (Riemann-Integral) 9. Approximation von Nullstellen und Fixpunkten. Das Newton-Verfahren	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden <ul style="list-style-type: none"> <li>• beschreiben den progressiven Aufbau des Zahlensystems (bis hin zu den komplexen Zahlen) und argumentieren mit dem Permanenzprinzip als formaler Leitidee</li> <li>• verwenden die Begriffe der Konvergenz von Folgen und Reihen sowie der Vollständigkeit der reellen Zahlen formal sicher und erläutern diese Begriffe an tragenden Beispielen</li> <li>• beschreiben die Begriffe Stetigkeit und Differenzierbarkeit anschaulich und formal und begründen zentrale Aussagen über stetige und differenzierbare Funktionen, verwenden die Idee der Approximation durch Potenzreihen zur Beschreibung von Funktionen</li> <li>• definieren den Begriff des Integrals formal und verwenden ihn in mathematischen Zusammenhängen, interpretieren das Integrieren als Flächenmessung und als Mittelwertbildung,</li> <li>• erläutern und begründen den Hauptsatz der Differential- und Integralrechnung</li> <li>• nutzen Software zur Darstellung und Exploration mathematischer Modellierungen und als heuristisches Werkzeug zur Lösung von Anwendungsproblemen</li> </ul>	

<ul style="list-style-type: none"> <li>• kennen und reflektieren Fragen der Umsetzung numerischer Verfahren auf dem Computer (z.B. Komplexität, Genauigkeit)</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Analysis für Informatiker: Die Studierenden <ul style="list-style-type: none"> <li>• präsentieren und erklären mathematische Sachverhalte</li> <li>• denken konzeptionell, analytisch und logisch</li> <li>• denken und handeln eigenständig</li> <li>• erarbeiten sich interessen gelenkt selbstständig mathematische Einsichten</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Klausur (120 - 180 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 8 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Jürgen Klüners
<b>Lernmaterialien, Literaturangaben</b>
Analysis für Informatiker: keine

<b>Sonstige Hinweise</b>
keine

### 3.2 Wahlpflichtmodul: Angriffssicherer Softwareentwurf

Modulname	Angriffssicherer Softwareentwurf / Secure Software Engineering
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>Angriffssicherer Softwareentwurf : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Angriffssicherer Softwareentwurf: Vorlesung ( 45h / 105h / EN / WS / 100 ) Angriffssicherer Softwareentwurf: Übung ( 30h / 0h / EN / WS / 100 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Angriffssicherer Softwareentwurf: Es wird vorausgesetzt, dass die Teilnehmer den ersten Studienabschnitt des Bachelors abgeschlossen oder nahezu abgeschlossen haben. Hierüberhinaus gibt es keine besonderen Voraussetzungen für diese Veranstaltung.	
<b>Inhalte</b>	
Angriffssicherer Softwareentwurf: Was braucht es, um Softwaresysteme angriffssicher zu entwickeln? Dies ist die Schlüsselfrage, der wir in dieser Veranstaltung auf den Grund gehen. Um sie zu beantworten ist es erforderlich, ein Verständnis der folgenden Kernbereiche des angriffssicheren Softwareentwurfs zu entwickeln: Bedrohungsmodellierung, sicheres Design, sichere Programmierung, Sicherheitsvalidierung, sicheres Deployment und sichere Wartung. Diese Bereiche werden in dieser Veranstaltung auf beispielorientierte Weise abgedeckt. Diskutiert werden aktuelle Techniken, die auf diese Bereiche anwendbar sind, sowie die Lektionen, die aus konkreten Sicherheitsvorfällen gelernt werden können.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Ziel der Veranstaltung ist es, dass die Teilnehmer ein fundiertes Verständnis der allerwichtigsten Aspekte des angriffssicheren Softwareentwurfs erhalten. Das schließt die Fähigkeit ein, Bedrohungen von Softwaresystemen zu identifizieren und zu modellieren, um die gängigsten Klassen von Schwachstellen zu vermeiden, sowie Techniken und Werkzeuge zu identifizieren und anzuwenden, um das Einführen? von Sicherheitsschwachstellen zu verhindern oder zu identifizieren.	
<b>Nichtkognitive Kompetenzen</b>	



<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Angriffssicherer Softwareentwurf:
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eric Bodden
<b>Lernmaterialien, Literaturangaben</b>
Angriffssicherer Softwareentwurf: Empfohlenes Buch: Gary McGraw: Software Security: Building Security In (2006, Addison-Wesley Professional) Über UPB als e-book verfügbar
<b>Sonstige Hinweise</b>
Inhalte: 1. Einführung in die Bedrohungsmodellierung und -analyse 2. Buffer Overflows: Prinzip, Exploits und Gegenmaßnahmen 3. Andere Code Injection-Schwachstellen: Prinzipien, Exploits und Gegenmaßnahmen 4. Crypto: gängige Algorithmen und Fallstricke 5. Zugriffskontrolle in Java und Android 6. Informationsfluss und Nutzungskontrolle 7. Automatische Erkennung von Schwachstellen: Codeanalyse, Fuzz Testing, modellbasiertes Testen 8. Systematische Sicherheitsanalyse 9. Softwareaktualisierung und -wartung

### 3.3 Pflichtmodul: Bachelor-Abschlussarbeit

Modulname	Bachelor-Abschlussarbeit / Bachelor Thesis
Workload	450 h
Leistungspunkte	15 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Bachelorarbeit Abschlussarbeit : 6</li> <li>• Bachelorarbeit Arbeitsplan : 6</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
<p>Bachelorarbeit Abschlussarbeit: Kontaktzeiten, Ergebnispräsentation ( 15h / 345 h / DE / WS oder SS / 1 )</p> <p>Bachelorarbeit Arbeitsplan: Kontaktzeiten, Präsentation Arbeitsplan ( 15h / 75 h / DE / WS oder SS / 1 )</p>	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Erfolgreicher Abschluss aller Module des erstem Studienabschnitts sowie des Moduls Schlüsselqualifikation.	
<b>Empfohlene Kenntnisse</b>	
<p>Bachelorarbeit Abschlussarbeit: Je nach gewähltem Thema.</p> <p>Bachelorarbeit Arbeitsplan: Je nach gewähltem Thema.</p>	
<b>Inhalte</b>	
<p>Bachelorarbeit Abschlussarbeit: Eine Bachelorarbeit umfasst die Bearbeitung eines Themas mit schriftlicher Ausarbeitung und einer mündlicher Präsentation der Ergebnisse. Der Studierende soll zeigen, dass er innerhalb einer vorgegebenen Frist ein Thema der Informatik auf der Grundlage wissenschaftlicher Methoden bearbeiten kann. Die Aufgabe einer Bachelorarbeit kann beispielsweise die Entwicklung von Software, Hardware, eine Beweisführung oder eine Literaturrecherche umfassen.</p> <p>Bachelorarbeit 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.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Im Rahmen ihrer Abschlussarbeit bearbeiten die Studierenden ein Problem nach wissenschaftlichen Methoden innerhalb einer bestimmten Frist. Die im Zuge des Studiums erworbenen fachlich-methodischen sowie fachübergreifenden Kompetenzen sollen dazu entsprechend eingesetzt werden. Dazu gehören insbesondere auch die Strukturierung und Planung der einzelnen Arbeitsschritte sowie die Präsentation der Ergebnisse nach Abschluss der Arbeit.	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</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>Bachelorarbeit Abschlussarbeit: Selbständiges Arbeiten unterstützt durch individuelle Betreuung</p> <p>Bachelorarbeit Arbeitsplan: Direkte Absprache mit Betreuer.</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 48 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>Bachelorarbeit Abschlussarbeit: Je nach gewähltem Thema.</p> <p>Bachelorarbeit Arbeitsplan: Je nach gewähltem Thema.</p>
<b>Sonstige Hinweise</b>
Die Bachelorarbeit 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. Die Aufgabenstellung soll so gestaltet werden, dass sie einem Arbeitsaufwand von neun

Wochen Vollzeitarbeit entspricht. Die Arbeit wird studienbegleitend erstellt und muss fünf Monate nach der Ausgabe abgegeben werden. Sie soll einen Umfang von in der Regel nicht mehr als 60 DIN A4-Seiten haben.

### 3.4 Pflichtmodul: Berechenbarkeit und Komplexität

Modulname	Berechenbarkeit und Komplexität / Computability and Complexity
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Berechenbarkeit und Komplexität : 3</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Berechenbarkeit und Komplexität: Vorlesung ( 45h / 105h / DE / WS / 200 ) Berechenbarkeit und Komplexität: Übung ( 30h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Erfolgreicher Abschluss der Module Modellierung und Datenstrukturen und Algorithmen	
<b>Empfohlene Kenntnisse</b>	
Berechenbarkeit und Komplexität: Inhalte der Vorlesungen Modellierung sowie Datenstrukturen und Algorithmen.	
<b>Inhalte</b>	
Berechenbarkeit und Komplexität: Einführung in grundlegende Methoden und Techniken zur Charakterisierung der Schwierigkeit von Berechnungsproblemen. Als formales Rechenmodell werden Turingmaschinen definiert. Ausgehend hiervon werden die wichtigsten Begriffe und Techniken der Berechenbarkeitstheorie (wie z.B. Entscheidbarkeit, Unentscheidbarkeit, Diagonalisierung, Reduktionen) und der Komplexitätstheorie (wie z.B. Zeitkomplexität, Klassen P und NP, NP-Vollständigkeit, polynomielle Reduktionen) definiert und erläutert.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende kennen wesentliche Konzepte und Methoden der Berechenbarkeitstheorie und der Komplexitätstheorie. Sie können selbständig Probleme analysieren und klassifizieren. Studierende können Hypothesen zur Komplexität von Problemen entwickeln und diese anschließend verifizieren oder falsifizieren und darauf aufbauend neue Hypothesen formulieren.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Berechenbarkeit und Komplexität: Die Vorlesung nutzt Tafelanschrieb und Folien sowie kleine Aufgaben für die Studierenden während der Vorlesung. Sie wird sowohl durch Tafelübung als auch durch Kleingruppentutorien begleitet. Studierende haben in den Kleingruppen Gelegenheit, Aufgaben in der	

Gruppe zu bearbeiten und Übungsblätter durch Tutoren benoten zu lassen.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Berechenbarkeit und Komplexität: Michael Sipser, Introduction to the Theory of Computation, Uwe Schöning, Theoretische Informatik - kurz gefasst; Foliensatz der VL; Übungsblätter.
<b>Sonstige Hinweise</b>
keine

### 3.5 Wahlpflichtmodul: Betriebssysteme

Modulname	Betriebssysteme / Operating Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Betriebssysteme : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Betriebssysteme: Vorlesung ( 45h / 105h / EN / WS / 30 ) Betriebssysteme: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Betriebssysteme: Vorlesung Systemsoftware und systemnahe Programmierung.	
<b>Inhalte</b>	
Betriebssysteme: Im Rahmen der Veranstaltung werden grundlegende Konzepte von Betriebssystemen besprochen, sowie spezifische Eigenschaften von Echtzeitbetriebssystemen und Betriebssystemen für eingebettete Systeme. Themen: <ul style="list-style-type: none"> <li>• Parallelismus</li> <li>• Scheduling</li> <li>• Synchronisation</li> <li>• Inter-Process Communication</li> <li>• Memory Management</li> <li>• Security</li> <li>• Eingebettete Systeme</li> <li>• Echtzeitsysteme</li> </ul>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Lernziel ist das Verständnis fundamentaler Konzepte von Betriebssystemen. Die Studierenden verstehen diese Konzepte und sind in der Lage, diese an Beispielen anzuwenden.	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Betriebssysteme: Vorlesung mit praktischen Übungen
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 Minuten) oder 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Falko Dressler
<b>Lernmaterialien, Literaturangaben</b>
Betriebssysteme: Folien, Lehrbücher
<b>Sonstige Hinweise</b>
keine



### 3.6 Wahlpflichtmodul: Computer Graphics Rendering

Modulname	Computer Graphics Rendering / Computer Graphics Rendering
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Computer Graphics Rendering : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Computer Graphics Rendering: Vorlesung ( 45h / 105h / DE / WS / 90 ) Computer Graphics Rendering: Übung ( 30h / 0h / DE / WS / 20 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Computer Graphics Rendering: Grundkenntnisse in Linearer Algebra und Vektorrechnung, sowie ein sattelfeste Programmierausbildung, werden vorausgesetzt.	
<b>Inhalte</b>	
<p>Computer Graphics Rendering: Computergrafik wird oft als übergeordneter Begriff verwendet, um die Erzeugung und Manipulation von digitalen Bildern zu beschreiben. Sie ist das Fachgebiet, welche visuelle Kommunikation durch Berechnung ermöglicht. In diesem Modul geht es konkret um die Generierung von digitalen Bildern und Bildsequenzen aus (mathematisch beschriebenen) 3D Szenen. Dieser Prozess wird Rendering genannt. Durch moderne Hardware und neue informatische Methoden unterstützt, wird Echtzeit-Rendering immer komplexerer 3D Szenen möglich. Um Studierende auf diesen Weg zu führen, werden folgende Themen bearbeitet:</p> <ul style="list-style-type: none"> <li>• Geometrische Modellierung einer 3D Szene durch mathematische Beschreibungen, z.B. Punkte, Ebenen, Vektoren, Polyeder, oder gekrümmte Flächen.</li> <li>• Die moderne Rendering Pipeline mit Transformationen (Translation, Skalierung, Rotation, Projektion), lokaler Reflektion und Schattierung, Sichtbarkeit, Rasterung, Texturen und Anti-aliasing.</li> <li>• Fortgeschrittene Rendering Verfahren wie Scene Graph, Echtzeit-Schattenalgorithmen, Bildbasiertes Rendering (Image-Based Rendering), globale Reflexion, inkl. rekursives Raytracing, Radiosity, und andere Näherungen der Rendering Gleichung, Non-Photorealistic Rendering, oder Partikel Systeme. Eine moderne Shader-basierte API wird die Vorstellung der Algorithmen begleiten und den Studierenden Erfahrungen mit GPU Architekturen ermöglichen.</li> </ul>	

<b>Lernergebnisse / Fachkompetenzen</b>
Studierende benennen und erklären relevante Verfahren entlang der Rendering Pipeline. Sie beherrschen entsprechende Rechentechniken um wichtige Verfahren (z.B. Transformation, Projektion, Cohen-Sutherland Clipping, Culling, Beleuchtungsmodelle, Gouraud-Schattierung, Rasterung von Linien und Kreisen, mathematische Faltung (convolution), B-Splines) auch in Rechenschritten nachzuvollziehen. Dasselbe (erklären, beherrschen Rechentechniken) gilt für alternative Verfahren zur Rendering Pipeline (z.B. Raytracing, Radiosity). Studierende demonstrieren die Fähigkeit, mit einem modernen API (z.B. OpenGL, WebGL) 3D Szenen nach bestimmten Vorgaben (Kamera, Beleuchtung, Modelle) mit unterschiedlichen Rendering Effekten (z.B. Schattenwurf, Bump Mapping, Environment Mapping) umzusetzen. Sie entwickeln auch Grafikprogrammen, welche die GPUs optimal ausnutzen. Studierende sind in der Lage, Rendering Software in Bezug auf Ihre Mächtigkeit an Rendering Funktionen zu bewerten.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Computer Graphics Rendering: Die Vorlesung nutzt Beamer und Tafel wechselweise. Die Studierenden bearbeiten kurze In-Class Aufgaben und diskutieren dann mit der Dozentin über unterschiedliche Lösungen bzw. Probleme bei den Lösungen. In den Übungen werden in Kleingruppen Aufgaben (mathematische Aufgaben, Algorithmen, Programmieraufgaben) bearbeitet und Hausaufgaben vorbereitet und nachbesprochen.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gitta Domik-Kienegger
<b>Lernmaterialien, Literaturangaben</b>
Computer Graphics Rendering: Vorlesungsfolien; Textbücher werden in der Vorlesung vorgestellt. Verteilung der Materialien über koaLA. Da die Computergrafik ein sehr dynamisches Fachgebiet ist, ändern sich die Materialien zeitlich entsprechend. Zur Zeit der Erstellung des Modulhandbuchs ist das Textbuch "Interactive Computer Graphics" von E. Angel und D. Schreiner, 6. Edition, Pearson Verlag, in Gebrauch.
<b>Sonstige Hinweise</b>
keine

### 3.7 Wahlpflichtmodul: Data Mining

Modulname	Data Mining / Data Mining
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Data Mining : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Data Mining: Vorlesung ( 45h / 105h / DE / SS / 120 ) Data Mining: Übung ( 30h / 0h / DE / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Data Mining: Grundlegende Kenntnisse in Mathematik (lineare Algebra, Statistik), Programmierung, Algorithmen und Datenstrukturen.	
<b>Inhalte</b>	
Data Mining: Die Vorlesung gibt eine Einführung in Verfahren zur Wissensentdeckung mithilfe von Methoden zur systematischen Suche nach Mustern in Daten. Der Schwerpunkt liegt auf effizienten algorithmischen Ansätzen für potenziell sehr große Datenbestände.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden verfügen über ein grundlegendes Verständnis der formalen statistischen und informationstheoretischen Grundlagen der Datenanalyse. Sie sind in der Lage, Datenanalyse- und Data Mining Probleme formal zu modellieren, Rohdaten eines speziellen Anwendungskontextes adäquat aufzubereiten, und geeignete Methoden auf die Daten anzuwenden. Darüber hinaus sind sie in der Lage, die Ergebnisse zu interpretieren und entsprechende Rückschlüsse zu ziehen. Insbesondere haben die Studierenden ein Bewusstsein für die Grenzen datenanalytischer Verfahren und die Gefahr von Fehlinterpretationen entwickelt.	
<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>
Data Mining: Theoretische Grundlagen und Konzepte des Data Mining werden im Rahmen einer Vorlesung eingeführt und anschließend in praktischen Übungen in Kleingruppen sowie in Heimübungen vertieft ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eyke Hüllermeier
<b>Lernmaterialien, Literaturangaben</b>
Data Mining: Begleitmaterial in Form eines Skripts, Übungszettel, zusätzliche Literatur in Form von Lehrbüchern.
<b>Sonstige Hinweise</b>
keine

### 3.8 Wahlpflichtmodul: Databases and Information Systems

Modulname	Databases and Information Systems / Databases and Information Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Databases and Information Systems : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Databases and Information Systems: Vorlesung ( 45h / 105h / EN / WS / 120 ) Databases and Information Systems: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Databases and Information Systems: Studierende sollten Vorkenntnisse in relationalen Datenbanken und SQL haben, die in etwa denen der Vorlesung "Datenbanksysteme" entsprechen, sowie Vorkenntnisse im Programmieren, die in etwa denen der Vorlesungen "Programmierung" und "Grundlagen der Programmiersprachen" entsprechen.	
<b>Inhalte</b>	
Databases and Information Systems: Datenspeicherung und Datenmanagement spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissens von Unternehmen in Daten abgelegt ist. Zukünftige Anwendungsentwickler brauchen zur Verarbeitung großer Datenmengen Kenntnisse, die über traditionelle Datenbanken hinausgehen, insbesondere über NoSQL und Nicht-Standard-Datenmodelle, Hauptspeicher-Datenbanken, Kompression, Indizierung und effiziente Suche, um Anwendungen oder Informationssysteme zu entwickeln, die akzeptable Antwortzeiten haben. Dieses Modul behandelt schwerpunktmäßig Anfragen und Verarbeitung von Massendaten einschließlich baumstrukturierter Daten, Textdaten, Datenströmen, NoSQL-Daten und Hauptspeicher-Datenbanken. Das Modul umfasst Algorithmen, Technologien und praktische Fertigkeiten in diesen Gebieten.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Nach Abschluss des Moduls sind die Studierenden in der Lage XML-Verarbeitung in Softwaresystemen zu verstehen, zu entwerfen, zu implementieren und in Bezug auf ihren Zeit- und Platz-Bedarf zu beurteilen. Sie kennen wesentliche Such- und Anfragetechniken zur Informationsbeschaffung in unkom-	

primierten oder komprimierten XML-Datenbeständen. Sie sind in der Lage, unendliche Datenströme geeignet zu verarbeiten. Die Studenten sind in der Lage, sich neueste Forschungsergebnisse anhand von wissenschaftlichen Papers zu erarbeiten.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Databases and Information Systems: Grundlegende Konzepte werden in einer Vorlesung präsentiert. Zusätzlich werden theoretische Konzepte in Tutorien in Kleingruppen vertieft, insbesondere für Kernkonzepte von Datenbanken wie die Suche und Anfragen auf Big Data, verteilten Datenbanken und mobilem Datenmanagement. Zudem erwerben Studierende praktische Kenntnisse durch Computergestützte Übungen, in denen sie aufbauend auf den in der Vorlesung erläuterten Konzepten ihre eigenen Informationssysteme, Such- oder Kompressionsalgorithmen entwickeln.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Stefan Böttcher
<b>Lernmaterialien, Literaturangaben</b>

Databases and Information Systems: Verweise auf aktuelles Lehrmaterial werden in der Vorlesung gegeben
<b>Sonstige Hinweise</b>
keine



### 3.9 Pflichtmodul: Datenbanksysteme

Modulname	Datenbanksysteme /
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Datenbanksysteme : 2</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Datenbanksysteme: Vorlesung ( 30h / 90h / DE / SS / 400 ) Datenbanksysteme: Übung ( 30h / 0h / DE / SS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Datenbanksysteme: Kenntnisse in der Programmierung werden in dem Umfang vorausgesetzt, wie sie in den Veranstaltungen Programmierung und GPS gelehrt werden. Elementare Kenntnisse der Logik der Modellierung aus der Vorlesung Modellierung werden ebenfalls vorausgesetzt.	
<b>Inhalte</b>	
<p>Datenbanksysteme: Datenbanken spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissen der Unternehmen als Daten in Datenbanken gespeichert wird. Für das Unternehmen ist es entscheidend, dass diese Daten korrekt, insbesondere konsistent, sind und dass sie effizient erfragt und aktualisiert werden können. Weiterhin sind die in Datenbanken abgelegten Datenbestände die wesentliche Datenquelle für eine Vielzahl von Anwendungsprogrammen, sie werden aber auch durch Anwendungsprogramme aktualisiert. Deshalb kommt der Organisation und Verarbeitung großer Datenbestände sowie der Einbindung von Datenbanken in Anwendungen eine zentrale Rolle bei der Erstellung korrekter und effizienter Anwendungen zu. Dieses Modul erschließt die Grundlagen für Datenbanksysteme, die in nahezu allen Unternehmen in der Praxis eingesetzt werden.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Studierende lernen Faktenwissen</p> <ul style="list-style-type: none"> <li>• Theorie und Konzepte relationaler Anfragesprachen kennen</li> <li>• Konzepte des Datenbankentwurfs kennen</li> <li>• Konzepte der Synchronisation und Recovery von Transaktionen kennen</li> </ul> <p>Vermittlung von methodischem Wissen in Kleingruppen-Präsenz-Übungen:</p> <ul style="list-style-type: none"> <li>• Komplexe Anfragen an relationale Datenbanken korrekt zu formulieren</li> <li>• ein Datenbankschema möglichst redundanzfrei zu entwerfen</li> </ul> <p>in praktischen Übungen am Rechner:</p> <ul style="list-style-type: none"> <li>• eigene SQL-Anfragen an existierende relationale Datenbanken stellen</li> </ul>	

<ul style="list-style-type: none"> <li>• Programme zu schreiben, die Datenbestände aus Datenbanken lesen oder verändern</li> <li>• eigene Datenbanken zu definieren und aufzubauen</li> </ul> <p>Vermittlung von Transferkompetenz</p> <ul style="list-style-type: none"> <li>• die erworbenen Kompetenzen und Fertigkeiten auf andere Datenquellen oder andere Datenbanksysteme zu übertragen</li> <li>• Umgang mit Zugriffsrechten</li> </ul> <p>Vermittlung von normativ-bewertenden Kompetenzen</p> <ul style="list-style-type: none"> <li>• die Eignung und Grenzen des relationalen Datenmodells bewerten und einzuschätzen</li> <li>• den Programmieraufwand für Datenbankabfragen und Datenbankprogrammierung einzuschätzen</li> <li>• die Folgen einer Datenbankschema-Änderung zu erkennen und abzuschätzen</li> <li>• die Risiken eines schlecht entworfenen Datenbankschemas zu bewerten</li> <li>• den Aufwand und Nutzen von Synchronisation und Recovery abzuschätzen</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Datenbanksysteme: Die Grundlagen und Konzepte von Datenbanksystemen werden im Rahmen einer Vorlesung eingeführt und anschließend in Präsenzübungen in Kleingruppen sowie in Heimübungen vertieft und durch praktische Übungen ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 5 Credits gewichtet.

<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Stefan Böttcher
<b>Lernmaterialien, Literaturangaben</b>
Datenbanksysteme: Lehrbuch: Kemper, Eickler: Datenbanksysteme , Oldenbourg-Verlag, neueste Ausgabe. Lehrbuch: Garcia-Molina, Ullman, Widom: Database Systems: The Complete Book, Prentice Hall, neueste Ausgabe.
<b>Sonstige Hinweise</b>
keine

### 3.10 Pflichtmodul: Datenstrukturen und Algorithmen

Modulname	Datenstrukturen und Algorithmen / Data Structures and Algorithms
Workload	270 h
Leistungspunkte	9 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Datenstrukturen und Algorithmen : 2</li> <li>• Praktikum: Datenstrukturen und Algorithmen : 2</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Datenstrukturen und Algorithmen: Vorlesung ( 60h / 135h / DE / SS / 400 ) Datenstrukturen und Algorithmen: Übung ( 30h / 0h / DE / SS / 25 ) Datenstrukturen und Algorithmen: Zentralübung ( 15h / 0 h / DE / SS / 400 ) Praktikum: Datenstrukturen und Algorithmen: ( 0h / 30h / DE / SS / 3 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Datenstrukturen und Algorithmen: Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse u.a. mit mathematischen Methoden zu erlernen. Praktikum: Datenstrukturen und Algorithmen: Grundlagen der Programmierung.	
<b>Inhalte</b>	
<p>Datenstrukturen und Algorithmen: Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus "für den Rechner verstehbar". Algorithmen spielen daher eine zentrale Rolle in der Informatik. Wesentliches Ziel des Algorithmenentwurfs ist die (Ressourcen-) Effizienz, d.h. die Entwicklung von Algorithmen, die ein gegebenes Problem möglichst schnell oder mit möglichst geringem Speicherbedarf lösen. Untrennbar verbunden mit effizienten Algorithmen sind effiziente Datenstrukturen, also Methoden, große Datenmengen im Rechner so zu organisieren, dass Anfragen wie Suchen Einfügen, Löschen aber auch komplexere Anfragen effizient beantwortet werden können. Die in dieser Veranstaltung vorgestellten Entwurfs- und Analysemethoden für effiziente Algorithmen und Datenstrukturen, sowie die grundlegenden Beispiele wie Sortierverfahren, dynamische Suchstrukturen und Graphenalgorithmen gehören zu den wissenschaftlichen Grundlagen für Algorithmenentwicklung und Programmierung in weiten Bereichen der Informatik.</p> <p>Praktikum: Datenstrukturen und Algorithmen: Begleitend zur Vorlesung Datenstrukturen und Algorithmen werden in diesem Programmierpraktikum einige wichtige Algorithmen und Datenstrukturen exemplarisch implementiert. Studierende werden in konkreten Projekten das Problem analysieren,</p>	

geeignete Programmiertechniken auswählen, praktisch realisieren und eine quantitative Leistungsbeurteilung durchführen.
<b>Lernergebnisse / Fachkompetenzen</b>
Die Studierenden kennen effiziente Datenstrukturen und Algorithmen für ausgewählte grundlegende Probleme. Sie sind in der Lage Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen einzusetzen. Sie können selbstständig und kreative Algorithmen und Datenstrukturen entwickeln (Wie gestalte ich den kreativen Prozess vom algorithmischen Problem zum effizienten Algorithmus?). Sie sind in der Lage mathematischer Methoden zum Korrektheitsbeweis und zur Effizienzanalyse einzusetzen. Sie können die Wechselwirkung zwischen Algorithmus und Datenstruktur an wesentlichen Beispielen erläutern. Sie können die Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten einschätzen. Sie können sich neue Algorithmen, Datenstrukturen und algorithmischen Ideen und Analysen aneignen.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Haltung und Einstellung</li> <li>• Lernkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Datenstrukturen und Algorithmen:</p> <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> <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> <p>Praktikum: Datenstrukturen und Algorithmen: Programmieraufgaben in kleineren Teams.</p>
<b>Prüfungsleistung (Dauer)</b>
<p>Klausur (120 - 180 Minuten)</p> <p>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</p> <p>Qualifizierte Teilnahme: Praktikumsarbeit mit anschließendem Gespräch</p> <p>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 9 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>
Datenstrukturen und Algorithmen: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter Praktikum: Datenstrukturen und Algorithmen: Aufgabenstellung, man pages, eigenständige Recherche zu weiterführender Literatur.
<b>Sonstige Hinweise</b>
keine

### 3.11 Pflichtmodul: Digitaltechnik

Modulname	Digitaltechnik / Digital Design
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Digitaltechnik : 2</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Digitaltechnik: Vorlesung ( 30h / 90h / DE / SS / 300 ) Digitaltechnik: Übung ( 30h / 0h / DE / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Digitaltechnik: Kenntnisse aus der Lehrveranstaltung Modellierung sind hilfreich.	
<b>Inhalte</b>	
Digitaltechnik: Die Vorlesung gibt eine Einführung in den Entwurf digitaler Schaltungen und Systeme. Dabei wird der Bogen vom Logikentwurf auf Gatterebene bis hin zu komplexeren Systemen auf Register-Transfer-Ebene gespannt. Die vermittelten Techniken und Methoden werden in den Übungen an Beispielen vertieft und mit modernen Entwurfswerkzeugen umgesetzt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• den Entwurfsablauf in der Digitaltechnik von der Spezifikation bis zur technischen Realisierung zu beschreiben,</li> <li>• die zugrunde liegenden mathematischen Modelle aus der Booleschen Algebra und der Automatentheorie zu erklären und anzuwenden,</li> <li>• Entwürfe im Hinblick auf vorgegebene Entwurfsziele zu analysieren und zu bewerten, sowie</li> <li>• einfache Systeme selbständig zu konzipieren und mit den entsprechenden Entwurfswerkzeugen technisch zu realisieren.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Digitaltechnik: <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> <li>• Rechnerübungen zum Thema Hardware-Entwurf (Teamarbeit)</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Klausur (60 - 90 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 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Digitaltechnik: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• J. F. Wakerly, "Digital Design," 4th Edition, Upper Saddle River, NJ: Pearson, Prentice Hall, 2007</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



### 3.12 Wahlpflichtmodul: Einführung in Kryptographie

Modulname	Einführung in Kryptographie / Introduction to Cryptography
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Einführung in Kryptographie : 6</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Einführung in Kryptographie: Vorlesung ( 45h / 105h / DE / SS / 100 ) Einführung in Kryptographie: Übung ( 30h / 0h / DE / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Einführung in Kryptographie: Datenstrukturen und Algorithmen sowie Berechenbarkeit und Komplexität	
<b>Inhalte</b>	
Einführung in Kryptographie: In dieser Vorlesung werden die wichtigsten Aufgaben und Methoden der modernen Kryptographie vorgestellt. Weiter werden einige der wichtigsten Sicherheitsanforderungen moderner Kryptographie informell diskutiert. Es werden die Vor- und Nachteile symmetrischer und asymmetrischer Kryptographie erläutert. Wichtige kryptographische Basiskonstruktionen wie Verschlüsselungsverfahren und digitale Signaturen werden vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende sind in der Lage Sicherheitsanforderungen von Anwendungen und Kryptosystemen präzise zu formulieren. Sie beherrschen die wichtigsten kryptographischen Basistechniken und können ihre Einsatzmöglichkeiten einschätzen und erläutern. Studierende können einschätzen, ob umgesetzte kryptographische Lösungen gegebenen Anforderungen genügen und sie können für gegebene Sicherheitsanforderungen die geeigneten kryptographischen Verfahren auszuwählen. Studierende können einschätzen, welche Anpassungen an kryptographische Verfahren unproblematisch sind und welche sicherheitskritisch sind.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> </ul>	

<ul style="list-style-type: none"> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Einführung in Kryptographie: 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>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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Einführung in Kryptographie: Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography, Chapman and Hall, Johannes Buchmann: Einführung in Kryptographie, Springer Verlag, Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.13 Wahlpflichtmodul: Eingebettete Systeme

Modulname	Eingebettete Systeme / Embedded Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Eingebettete Systeme : 4</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Eingebettete Systeme: Vorlesung ( 45h / 105h / DE / SS / 50 ) Eingebettete Systeme: Übung ( 30h / 0h / DE / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Eingebettete Systeme: Kenntnisse aus der Lehrveranstaltung Rechnerarchitektur sind hilfreich.	
<b>Inhalte</b>	
Eingebettete Systeme: Die Veranstaltung bietet eine Einführung in Eingebettete Systeme und vermittelt Grundlagen zu Spezifikationsmodellen, eingebetteten Zielarchitekturen, grundlegenden Syntheseverfahren für Software und Hardware, sowie Methoden für die Bewertung und Analyse von Prozessor-Performance und -Energie.	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,</p> <ul style="list-style-type: none"> <li>• die Eigenschaften eingebetteter Systeme zu benennen,</li> <li>• die Entwurfsziele und Eigenschaften wesentlicher Typen von eingebetteten Zielarchitekturen zu erklären,</li> <li>• Methoden der Codegenerierung und -optimierung für eingebettete Prozessoren anzugeben,</li> <li>• die Zusammenhänge an der Hardware/Software-Grenze zwischen Codegenerator und Prozessorarchitektur zu beschreiben,</li> <li>• grundlegende Verfahren von Software- und Hardware-Synthese zu erklären,</li> <li>• die Bedeutung von Performance- und Energie-Metriken einzuschätzen und</li> <li>• Methoden zur Bestimmung der Worst-Case-Execution-Time anzuwenden.</li> </ul>	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Eingebettete Systeme: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Interaktive Übungen im Hörsaal</li> <li>• Rechnerübungen mit eingebetteten Zielarchitekturen (DSP, ARM, FPGA)</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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Eingebettete Systeme: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Aufgabenblätter und technische Dokumentation für die Rechnerübungen</li> <li>• Peter Marwedel: Embedded System Design, Springer, 2011.</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

### 3.14 Pflichtmodul: Gestaltung von Nutzungsschnittstellen

Modulname	Gestaltung von Nutzungsschnittstellen / Designing User Interfaces
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Gestaltung von Nutzungsschnittstellen : 4</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Gestaltung von Nutzungsschnittstellen: Vorlesung ( 45h / 105h / DE / SS / 300 ) Gestaltung von Nutzungsschnittstellen: Übung ( 30h / 0h / DE / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Gestaltung von Nutzungsschnittstellen:	
<b>Inhalte</b>	
<p>Gestaltung von Nutzungsschnittstellen: Erlernbarkeit, Beeinträchtigungsfreiheit sowie die barrierefreie Erschließbarkeit von Softwaresystemen sind heute Pflichtanforderungen an die Systemgestaltung. Grundlegende Herausforderungen sind, mit Hilfe geeigneter Gestaltungsmaßnahmen Verständnisprozesse bei Nutzern zu fördern und unnötige Belastungen bei der Arbeit mit Softwaresystemen zu vermeiden. Dazu ist ein methodisches Repertoire erforderlich, um schon während des Entwurfs die Gebrauchstauglichkeit sichern zu können. Die dazu erforderlichen Kenntnisse und Fertigkeiten reichen von physiologischen und psychologischen Grundlagen über Methoden und Techniken der Systemgestaltung bis hin zu rechtlichen Anforderungen.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Die Studierenden sind in der Lage, grundlegende Problembereiche der Mensch-Rechner-Interaktion zu erkennen und sie konstruktiv gestaltend umzusetzen. Sie erwerben zugleich anschlussfähiges Wissen, das vor allem für die Zusammenarbeit mit Designern und Psychologen erforderlich ist, aber auch für den Diskurs mit Medienwissenschaftlern und Pädagogen (E-Learning) hilfreich ist. Die erworbenen Kenntnisse und Fertigkeiten bilden zugleich die Grundlage für vertiefende Veranstaltungen im Bereich der Mensch-Maschine-Wechselwirkung wie etwa Usability Engineering, Computergrafik oder auch Medien-Ergonomie, d.h. die Teilnehmer können Gebrauchstauglichkeit evaluieren, graphische Darstellungen erzeugen und manipulieren, sowie digitale Medien einordnen und bewerten.</p>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Medienkompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
Gestaltung von Nutzungsschnittstellen: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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.-Ing. Reinhard Keil
<b>Lernmaterialien, Literaturangaben</b>
Gestaltung von Nutzungsschnittstellen: Benyon, David (2010): Designing interactive systems. A comprehensive guide to HCI and interaction design. 2nd ed. Harlow: Addison Wesley. Leventhal, Laura M.; Barnes, Julie A. (2008): Usability engineering. Process, products, and examples. Upper Saddle River, NJ: Pearson/Prentice Hall. Nielsen, Jakob (1996): Usability engineering. 3. [print]. Boston: Acad. Press. Raskin, Jef (2009): The Humane Interface. New directions for designing interactive systems. 11. print. Boston, Mass.: Addison Wesley. Shneiderman, Ben; Plaisant, Catherine (2010): Designing the user interface. Strategies for effective human-computer interaction. 5. ed., internat. ed. Upper Saddle River, NJ: Addison-Wesley/Pearson. Tognazzini, Bruce (1992): Tog on interface. 2. printing. Reading, Mass.: Addison-Wesley.
<b>Sonstige Hinweise</b>
keine

### 3.15 Wahlpflichtmodul: Grundlagen Wissensbasierter Systeme

Modulname	Grundlagen Wissensbasierter Systeme / Foundations of Knowledge-Based Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Grundlagen Wissensbasierter Systeme : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Grundlagen Wissensbasierter Systeme: Vorlesung ( 45h / 105h / DE / WS / 120 ) Grundlagen Wissensbasierter Systeme: Übung ( 30h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Grundlagen Wissensbasierter Systeme:	
<b>Inhalte</b>	
Grundlagen Wissensbasierter Systeme: 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 wissenschaftlichen Ebene 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. Diese Vorlesung gibt eine Einführung in Methoden und Konzepte der Künstlichen Intelligenz. Der inhaltliche Schwerpunkt liegt dabei auf Wissensbasierten Systemen im Sinne von Systemen, die mithilfe adäquater Ansätze zur Repräsentation und Verarbeitung von Wissen die Problemlösungskompetenz eines Fachexperten in einer bestimmten Anwendungsdomäne approximieren. Neben Methoden der Wissensrepräsentation und -verarbeitung gibt die Vorlesung auch einen ersten Einblick in den automatisierten Erwerb von Wissen mithilfe maschineller Lernverfahren.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden verstehen den Unterschied zwischen klassischen Softwaresystemen und wissensbasierten Systemen bzw. klassischer Programmierung und dem Entwurf wissensbasierter Systeme. Sie sind mit der Architektur wissensbasierter Systeme sowie grundlegenden Methoden und Techniken zum Entwurf solcher Systeme vertraut und können sie auf konkrete Probleme anwenden. Die Studierenden	



verstehen das Zusammenspiel von Wissen, Daten und Inferenz.
<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>
Grundlagen Wissensbasierter Systeme: Theoretische Grundlagen und Konzepte der Künstlichen Intelligenz werden im Rahmen einer Vorlesung eingeführt und anschließend in praktischen Übungen in Kleingruppen sowie in Heimübungen vertieft ergänzt.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eyke Hüllermeier
<b>Lernmaterialien, Literaturangaben</b>
Grundlagen Wissensbasierter Systeme: Skript und weitere Literatur in Form von Lehrbüchern.
<b>Sonstige Hinweise</b>
keine

### 3.16 Wahlpflichtmodul: Grundlegende Algorithmen

Modulname	Grundlegende Algorithmen / Fundamental Algorithms
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Grundlegende Algorithmen : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Grundlegende Algorithmen: Vorlesung ( 45h / 105h / DE / WS / 100 ) Grundlegende Algorithmen: Übung ( 30h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Grundlegende Algorithmen: 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>	
Grundlegende Algorithmen: Dieser Kurs präsentiert Algorithmen und algorithmische Paradigmen für grundlegenden Probleme. Paradigmen wie Teile und Herrsche, dynamische Programmierung und Greedy-Algorithmen werden erläutert und durch Beispiele veranschaulicht. Ferner werden Graphenalgorithmen, Netzwerk-Fluss-Algorithmen und Hash-Verfahren vorgestellt. In allen Fällen werden Korrektheitsbeweise und Laufzeitanalysen angegeben.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden wenden Entwurfsmethoden für effiziente Datenstrukturen und Algorithmen für schwierige Probleme wie Matching, Netzwerk-Fluß u.a. an. Sie nutzen mathematisch fundierte Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen. Zudem wählen sie zielgerichtet die zum gegebenen Problem passende generische Entwurfsmethode - z.B. Greedy-Algorithmen, Divide and Conquer, Dynamische Programmierung u.v.a.- aus. Darüber hinaus entwickeln sie selbstständig, kreativ Algorithmen und Datenstrukturen (Wie gestalte ich den kreativen Prozess vom algorithmischen Problem zum effizienten Algorithmus?) unter Nutzung von Entwurfsmethoden und ihrem Verständnis für die Struktur des algorithmischen Problems. Zudem nutzen sie einfache Varianten von fortgeschrittenen algorithmische Modellen wie online, approximative oder randomisierte Algorithmen.	

men.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Haltung und Einstellung</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Grundlegende Algorithmen:</p> <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> <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>
<p>Klausur (90 - 120 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 12 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>
Grundlegende Algorithmen: Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter

<b>Sonstige Hinweise</b>
keine

### 3.17 Wahlpflichtmodul: Interaktionsgestaltung

Modulname	Interaktionsgestaltung / Interaction Design
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Interaktionsgestaltung : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Interaktionsgestaltung: Vorlesung ( 45h / 105h / DE / WS / 70 ) Interaktionsgestaltung: Übung ( 30h / 0h / DE / WS / 70 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Interaktionsgestaltung: Die Vorlesung baut auf der vorangegangenen Pflichtvorlesung Gestaltung von Nutzungsschnittstellen auf, in der bereits grundlegende Inhalte des Usability Engineering eingeführt werden.	
<b>Inhalte</b>	
Interaktionsgestaltung: Die Vorlesung behandelt die Gestaltung von Interaktion mit klassischen, graphischen Desktopsystemen genauso wie die Interaktionsgestaltung im Web. Dabei baut sie auf der Pflichtvorlesung „Gestaltung von Nutzungsschnittstellen“ auf, in der erste Grundkenntnisse des Usability Engineering vermittelt wurden. Hier werden die Inhalte vertieft und besonders intensiv die existierenden Usability-Regeln, sowie die darauf aufbauenden Verfahren behandelt und auch praktisch eingeübt. Es wird anschließend deutlich gemacht, was sich ändert, wenn wir statt klassischer graphischer Oberflächen die Situation bei der Gestaltung von Webauftritten betrachten. Die deutlich existierenden Gemeinsamkeiten beider Bereiche, aber auch ihre Abgrenzungen werden deutlich gemacht. Im Web tritt an die Stelle des Dialogdesigns das Navigationsdesign, was alternative Entwicklungsstrategien erfordert. Besondere Betonung wird in der Vorlesung auf die stets wachsende Nutzung mobiler Geräte mit ihrer andersartigen technischen Ausrüstung gelegt. Dadurch und durch den sehr anderen Nutzungskontext entstehen neue Benutzungsanforderungen, die durch neuartige Entwicklungsprozesse und -techniken abgefangen werden müssen. Um das Thema abzurunden, wird schließlich die bedeutende Rolle von Farbe und Typographie im Web thematisiert.	
<b>Lernergebnisse / Fachkompetenzen</b>	

Die Studierenden lernen die wesentlichen, aktuellen Verfahren kennen und anwenden, die im klassischen Usability Engineering, aber auch im Web eine Rolle spielen. Die Studierenden lernen, sich in der umfangreichen und komplexen Welt von Usability-Regeln zu orientieren und diese anzuwenden. Bezogen auf die Gestaltung von Interaktion im Web lernen die Hörer der Vorlesung relevante Aspekte der Webgestaltung zu trennen und adäquat zu untersuchen: Inhaltsstrukturen, visuelle Anordnung, Navigationsstrukturen und Auswahl von Typographie und Farbe.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Interaktionsgestaltung: 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 und Videoaufzeichnungen aller Vorlesungen zur Verfügung gestellt. Im Rahmen der Vorlesung selbst findet immer wieder an geeigneter Stelle interaktive Gruppenarbeit statt – etwa zum Durchführen von Usability-Tests oder Anwendung von Inspektionsmethoden. 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>
Klausur (90 - 120 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 12 Credits gewichtet.

<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Gerd Szwillus
<b>Lernmaterialien, Literaturangaben</b>
Interaktionsgestaltung: <ul style="list-style-type: none"><li>• Vorlesungsfolien als PDF zum Herunterladen</li><li>• Schriftliche Hausaufgaben</li><li>• Diverse, während der Vorlesung eingesetzte Softwarewerkzeuge</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>• Jakob Nielsen und Raluca Budiu: Mobile Usability, 2012</li></ul>
<b>Sonstige Hinweise</b>
keine

### 3.18 Pflichtmodul: IT Sicherheit

Modulname	IT Sicherheit / IT Security
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• IT Sicherheit : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
IT Sicherheit: Vorlesung ( 30h / 90h / DE / WS / 150 ) IT Sicherheit: Übung ( 30h / 0h / DE / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Erfolgreicher Abschluss des Moduls Systemsoftware und systemnahe Programmierung	
<b>Empfohlene Kenntnisse</b>	
IT Sicherheit: Rechnernetze, Programmierung, Systemsoftware	
<b>Inhalte</b>	
IT Sicherheit: In der Vorlesung werden die wesentlichen Begriffe und Probleme der IT Sicherheit vorgestellt. Es werden klassische und moderne Angriffstechniken auf Netzwerkprotokolle, Passwort-Datenbanken, Computersysteme und Webanwendungen werden vorgestellt und geeignete Gegenmaßnahmen diskutiert. Hierzu gehört auch die Vorstellung praxisrelevanter kryptographischer Protokolle und Algorithmen, sowie deren Sicherheitseigenschaften.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende verstehen die wesentlichen Konzepte, Methoden und Mechanismen zum Schutz von Daten und Systemen vor Manipulation und Missbrauch auf einem grundlegenden, praxis-orientierten, wissenschaftlichen Niveau. Sie sind in der Lage, die Konzepte zur Erhöhung der Systemsicherheit korrekt einzusetzen, einfache Sicherheitsprotokolle zu entwickeln und diese zu bewerten. Sie verstehen die Ursachen von Sicherheits-Problemen heutiger Systeme, sind in der Lage, grundlegende Konzepte auch in neuen Anwendungskontexten einzusetzen und besitzen ein generelles Bewusstsein für mögliche Sicherheitsbedrohungen und Risiken.	
<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>	
IT Sicherheit:	



<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> <li>• Praktische Übungen zur IT Sicherheit</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Klausur (60 - 90 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 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr., Tibor Jager
<b>Lernmaterialien, Literaturangaben</b>
IT Sicherheit: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• Sicherheit und Kryptographie im Internet, Jörg Schwenk</li> <li>• Computer Security, William Stallings und Lawrie Brown</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.19 Wahlpflichtmodul: Komplexitätstheorie

Modulname	Komplexitätstheorie / Complexity Theory
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Komplexitätstheorie : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Komplexitätstheorie: Vorlesung ( 45h / 105h / DE / WS oder SS / 30 ) Komplexitätstheorie: Übung ( 30h / 0h / DE / WS oder SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Komplexitätstheorie: Berechenbarkeit und Komplexität	
<b>Inhalte</b>	
Komplexitätstheorie: Die Komplexitätstheorie ist eine wichtige Ergänzung der Theorie der Algorithmen. Ihr Ziel ist es zu verstehen, warum gewisse Berechnungsprobleme schwierig sind und diese anhand ihrer Schwierigkeit zu klassifizieren. Das bekannteste und wichtigste Beispiel ist die Theorie der NP-Vollständigkeit.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende können die Eigenschaften wesentlicher Komplexitätsklassen wie P, NP und PSPACE benennen und erläutern. Studierende können Probleme in geeignete Komplexitätsklassen einordnen. Studierende beherrschen die wichtigsten Techniken, um Probleme gemäß ihrer Komplexität vergleichen zu können (Reduktionen). Sie können diese Techniken auf neue Probleme anwenden.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	

Komplexitätstheorie: 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>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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Johannes Blömer
<b>Lernmaterialien, Literaturangaben</b>
Komplexitätstheorie: Michael Sipser, Introduction to the Theory of Computation, S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press, Vorlesungsfolien, Übungsaufgaben
<b>Sonstige Hinweise</b>
keine

### 3.20 Pflichtmodul: Lineare Algebra für Informatiker

Modulname	Lineare Algebra für Informatiker / Linear Algebra for Computer Science
Workload	240 h
Leistungspunkte	8 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Lineare Algebra für Informatiker : 2</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Lineare Algebra für Informatiker: Vorlesung ( 60h / 150h / DE / SS / 300 ) Lineare Algebra für Informatiker: Übung ( 30h / 0h / DE / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Lineare Algebra für Informatiker: Analysis für Informatiker	

<b>Inhalte</b>
<p>Lineare Algebra für Informatiker: Einführung in die Grundlagen der Linearen Algebra, die während des Informatikstudiums benötigt werden. Die Lineare Algebra thematisiert, auf unterschiedlichen begrifflichen Ebenen, praktisch und theoretisch das Lösen linearer Gleichungssysteme und darüber hinausgehend das Konzept der Linearität als universell einsetzbares mathematisches Lösungswerkzeug. Dessen Rolle für das weitere Studium liegt in der großen Bedeutung, welche die Linearisierung (oder Lineare Ap-proximation) für alle Sparten der Mathematik, für die mathematische Modellbildung und für die mathematischen Anwendungen hat.</p> <p>Das Modul besteht aus der Vorlesung Lineare Algebra (für Informatiker) Gliederung Lineare Algebra (V4+Ü2)</p> <ol style="list-style-type: none"> <li>1. Grundbegriffe</li> <li>2. Vektorräume</li> <li>3. lineare Abbildungen</li> <li>4. Basis</li> <li>5. Dimension</li> <li>6. Matrizen</li> <li>7. lineare Gleichungssysteme</li> <li>8. Determinanten</li> </ol>

<p>9. Eigenwerte</p> <p>10. charakteristisches Polynom</p> <p>11. Normalformenproblem</p>
<b>Lernergebnisse / Fachkompetenzen</b>
<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• verstehen und erläutern, wie abstrakte Vektorräume als koordinatenfreie Verallgemeinerung ein- bis dreidimensionaler Räume zustande kommen, und geben Beispiele aus der Mathematik und Anwendungsgebieten an, die in diesem konzeptionellen Rahmen verstanden werden können</li> <li>• begreifen lineare Abbildungen von Vektorräumen als strukturverträgliche Abbildungen und erläutern, wie lineare Gleichungssysteme koordinatenfrei durch sie beschrieben werden</li> <li>• verstehen den abstrakten Basis- und Dimensionsbegriff und erklären, wie dieser als Verallgemeinerung des naiven Koordinaten- und Dimensionsbegriff verstanden werden kann</li> <li>• stellen lineare Abbildungen durch Matrizen dar und begreifen diese als koordinatenabhängige Realisierung</li> <li>• verstehen und erläutern, wie sich die (eindeutige) Lösbarkeit solcher Gleichungssysteme charakterisieren lässt; lösen lineare Gleichungssysteme und erklären Lösungsverfahren</li> <li>• verstehen die Determinante als alternierende Multilinearform und erläutern sie anhand ihrer geometrischen Bedeutung; begreifen ihre Rolle für die Inversion von Matrizen und kennen die Verfahren zu ihrer Bestimmung</li> <li>• kennen den Begriff des Eigenwerts; verstehen und erklären das Normalformenproblem, kennen Kriterien für Diagonalisierbarkeit</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Lineare Algebra für Informatiker: Die Studierenden</p> <ul style="list-style-type: none"> <li>• reflektieren eigene Lernerfahrungen</li> <li>• präsentieren und erklären mathematische Sachverhalte</li> <li>• denken konzeptionell, analytisch und logisch</li> <li>• erarbeiten sich interessengeleitet selbständig neue Erkenntnisse</li> <li>• denken und handeln eigenständig</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
<p>Klausur (120 - 180 Minuten)</p> <p>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</p> <p>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt</p>

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 8 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Jürgen Klüners
<b>Lernmaterialien, Literaturangaben</b>
Lineare Algebra für Informatiker: keine
<b>Sonstige Hinweise</b>
keine

### 3.21 Wahlpflichtmodul: Logik und Deduktion

Modulname	Logik und Deduktion / Logic and Deduction
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Logik und Deduktion : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Logik und Deduktion: Vorlesung ( 45h / 105h / DE / SS / 60 ) Logik und Deduktion: Übung ( 30h / 0h / DE / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Logik und Deduktion: Die Bachelorvorlesung Modellierung.	
<b>Inhalte</b>	
Logik und Deduktion: In dieser Vorlesung werden Grundlagen der mathematischen Logik vermittelt, die für die Informatik relevant sind. Die Studierenden lernen wesentliche Eigenschaften der klassische Aussagen- und Prädikatenlogik kennen. Darüber hinaus werden neuere Logiken, wie Modallogik und Fuzzy Logik, und deren Anwendungen in der Informatik eingeführt. Für diese logischen Systeme untersuchen wir Syntax, Semantik und verschiedene Beweiskalküle.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden lernen verschiedene logische Systeme kennen. Sie kennen wesentliche Eigenschaften dieser Systeme und sind in der Lage mit unterschiedlichen Beweiskalkülen zu arbeiten. Die Studierenden erwerben die Fähigkeit zur Modellierung mit Hilfe der Logik.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	

Logik und Deduktion: In den Vorlesungen werden alle Themen behandelt. Während der Übungen werden die Studierenden ausführlich mit den neuen Themen arbeiten. Es gibt kein Computerpraktikum.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulelprüfungen</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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Lorijn van Rooijen
<b>Lernmaterialien, Literaturangaben</b>
Logik und Deduktion: Wird noch bekannt gegeben.
<b>Sonstige Hinweise</b>
keine



### 3.22 Wahlpflichtmodul: Modellbasierte Softwareentwicklung

Modulname	Modellbasierte Softwareentwicklung / Model-Based Software Development
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Modellbasierte Softwareentwicklung : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Modellbasierte Softwareentwicklung: Vorlesung ( 45h / 105h / EN / SS / 75 ) Modellbasierte Softwareentwicklung: Übung ( 30h / 0h / EN / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Modellierung, Programmierung und Software Engineering	
<b>Empfohlene Kenntnisse</b>	
Modellbasierte Softwareentwicklung: Modellierung, Programmierung und Software Engineering	
<b>Inhalte</b>	
<p>Modellbasierte Softwareentwicklung: In der modellbasierten Softwareentwicklung steht das Modell einer Software im Mittelpunkt. Es wird dabei nicht nur zu Dokumentationszwecken, sondern auch zur Entwicklung selbst verwendet (auch modellgetriebene Softwareentwicklung genannt). Übliche modellbasierte Techniken beinhalten unter anderem den Entwurf von Modellierungssprachen anhand von statischer und dynamischer Semantik sowie Metamodellierung sowie die Anwendung der Modelle in Form von Modelltransformationen, oder auch zum Model Checking oder für das Reverse Engineering von Softwarearchitekturen. Den Trend zur modellbasierten und modellgetriebenen Softwareentwicklung kann man sowohl in der Forschung, als auch in der Praxis beobachten und stellt daher eine wichtige Grundlage für die Ausbildung eines Softwareentwicklers dar.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Die Studierenden sollen grundlegende Verfahren zur Konstruktion großer Softwaresysteme kennen und ihre Anwendung beherrschen. Sie sollen die Vor- und Nachteile von Spezifikationstechniken erfahren, die Notwendigkeit von Design erkennen und Modelle zur Verbesserung der Softwarequalität einsetzen können. Unter anderem wird auf das Paradigma des „Model Driven Development“ eingegangen, das einen wesentlichen Produktivitäts- und Qualitätsgewinn bei der Softwareentwicklung verspricht.</p>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
Modellbasierte Softwareentwicklung: Vorlesung mit Beamer und praktische Rechnerübungen.
<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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Jun.-Prof.Dr. Anthony Anjorin
<b>Lernmaterialien, Literaturangaben</b>
Modellbasierte Softwareentwicklung: <ul style="list-style-type: none"> <li>• Völter, Stahl: Model-Driven Software Development: Technology, Engineering, Management (Wiley)</li> <li>• Ghezzi: Fundamentals of Software Engineering (Addison Wesley)</li> </ul>
<b>Sonstige Hinweise</b>
keine

### 3.23 Pflichtmodul: Modellierung

Modulname	Modellierung / Modelling
Workload	240 h
Leistungspunkte	8 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Modellierung : 1</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Modellierung: Vorlesung ( 60h / 150h / DE / WS / 500 ) Modellierung: Übung ( 30h / 0h / DE / WS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Modellierung: keine, Veranstaltung im 1. Semester	
<b>Inhalte</b>	
<p>Modellierung: Das Modellieren ist eine für das Fach Informatik typische Arbeitsmethode, die in allen Gebieten des Faches angewandt wird. Aufgaben, Probleme oder Strukturen werden untersucht und als Ganzes oder in Teilaspekten beschrieben, bevor sie durch den Entwurf von Software, Algorithmen, Daten und/oder Hardware gelöst bzw. implementiert werden. Mit der Modellierung eines Problems zeigt man, ob und wie es verstanden wurde. Damit ist sie Voraussetzung und Maßstab für die Lösung und sie liefert meist auch den Schlüssel für einen systematischen Entwurf. Als Ausdrucksmittel für die Modellierung steht ein breites Spektrum von Kalkülen und Notationen zur Verfügung. Sie sind spezifisch für unterschiedliche Arten von Problemen und Aufgaben. Deshalb werden in den verschiedenen Gebieten der Informatik unterschiedliche Modellierungsmethoden eingesetzt. In den entwurfsorientierten Gebieten (Softwaretechnik, Hardware-Entwurf) ist die Bedeutung der Modellierung und die Vielfalt der Methoden natürlich besonders stark ausgeprägt.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende kennen wesentliche Techniken zur Modellierung informatischer Probleme. Sie können für ein gegebenes Problem eine geeignete Modellierungstechnik auswählen und das Problem mit dieser Technik beschreiben. Sie können grundlegende Techniken erweitern und verfeinern, um so neuartige Probleme zu modellieren.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> </ul>	

<b>Methodische Umsetzung</b>
Modellierung: Die Vorlesung nutzt Tafelanschrieb und Folien sowie kleine Aufgaben für die Studierenden während der Vorlesung. Sie wird sowohl durch Tafelübung als auch durch Kleingruppentutorien begleitet. Studierende haben in den Kleingruppen Gelegenheit, Aufgaben in der Gruppe zu bearbeiten und Übungsblätter durch Tutoren benoten zu lassen.
<b>Prüfungsleistung (Dauer)</b>
Klausur (120 - 180 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 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>
Modellierung: Uwe Kastens, Hans Kleine Büning, Modellierung; Angelika Steger, Diskrete Strukturen; Foliensatz der Vorlesung; Übungsblätter.
<b>Sonstige Hinweise</b>
keine

### 3.24 Wahlpflichtmodul: Parallelität und Kommunikation

Modulname	Parallelität und Kommunikation / Parallelism and Communication
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Parallelität und Kommunikation : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Parallelität und Kommunikation: Vorlesung ( 45h / 105h / DE / WS / 60 ) Parallelität und Kommunikation: Übung ( 30h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Parallelität und Kommunikation: Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analysen wird vorausgesetzt.	
<b>Inhalte</b>	
Parallelität und Kommunikation: Die Vorlesung beschäftigt sich mit effizienten Methoden, Kommunikation zwischen Mitglieder eines Netzwerks zu realisieren. Solche Netzwerke können z.B. LANs, WANs, Peer-to-Peer Systeme, das Internet oder Parallelrechner sein. In der Vorlesung stellen wir verteilte Algorithmen vor, Kommunikation durch Routing im Netzwerk, durch Simulation des Kommunikationsgraphen auf dem Netzwerk und mit Hilfe globaler Variablen zu realisieren. Zudem werden effiziente Methoden zur Verwaltung von globalem Speicher in Netzwerken vorgestellt. Diese Algorithmen werden bezüglich Korrektheit und Effizienz analysiert.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden lernen die wichtigsten Techniken und Algorithmen im Bereich Netzwerkkommunikation kennen. Sie können entscheiden, in welcher Situation welcher Routing-Algorithmus geeignet ist. Sie können Routing-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>
Parallelität und Kommunikation: <ul style="list-style-type: none"> <li>• Vorlesung mit Beamer und Tafelanschrieb</li> <li>• Übungen in Kleingruppen</li> <li>• erwartete Aktivitäten der Studierenden: Lösung von Übungsaufgaben, Mitarbeit in den Übungen</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 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>
Parallelität und Kommunikation: Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Frank Thomson Leighton, M. Kaufmann Publishers, 1992, Skript, Foliensatz der Vorlesung, Übungsblätter
<b>Sonstige Hinweise</b>
keine

### 3.25 Pflichtmodul: Programmiersprachen

Modulname	Programmiersprachen / Programming Languages
Workload	120 h
Leistungspunkte	4 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Programmiersprachen : 1</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Programmiersprachen: Vorlesung ( 30h / 75h / DE / WS / 500 ) Programmiersprachen: Übung ( 15h / 0h / DE / WS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Programmiersprachen: Programmierung	
<b>Inhalte</b>	
Programmiersprachen: In Programmiersprachen werden Sprachkonstrukte, Spracheigenschaften und Programmierparadigmen im Vergleich und in Gegenüberstellung zu den im Modul Programmierung gelernten herausgearbeitet. Funktionale und logische Sprachkonstrukte und Programmierkonzepte werden auch praktisch an Beispielen in SML und Prolog erarbeitet.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden können Grundkonzepte von Programmier- und Anwendungssprachen erläutern. Sie können typische Eigenschaften nicht-imperativer Sprachen erklären. Sie sind in der Lage einfache Grammatiken, Typenspezifikationen und funktionale Programme zu entwickeln. Sie sind in der Lage praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben zu übertragen. Sie besitzen die Fähigkeit neue Programmier- und Anwendungssprachen selbstständig zu erlernen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> </ul>	
<b>Methodische Umsetzung</b>	
Programmiersprachen:	
<b>Prüfungsleistung (Dauer)</b>	
Klausur (60 - 90 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 4 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Stefan Böttcher
<b>Lernmaterialien, Literaturangaben</b>
Programmiersprachen: keine
<b>Sonstige Hinweise</b>
keine



### 3.26 Wahlpflichtmodul: Programmiersprachen und Übersetzer

Modulname	Programmiersprachen und Übersetzer / Programming Languages and Compilers
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Programmiersprachen und Übersetzer : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Programmiersprachen und Übersetzer: Vorlesung ( 45h / 105h / EN / WS / 120 ) Programmiersprachen und Übersetzer: Übung ( 30h / 0h / EN / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Programmiersprachen und Übersetzer: Aus der Vorlesung “Modellierung”: Endliche Automaten, kontext-freie Grammatiken. Aus der Vorlesung “Grundlagen der Programmiersprachen”: Spracheigenschaften, kontext-freie Grammatiken, Gültigkeitsbereiche, Datentypen.	
<b>Inhalte</b>	
Programmiersprachen und Übersetzer: Sprachen spielen in der Softwaretechnik vielfältige und wichtige Rollen: Als Programmiersprachen sind sie Ausdrucksmittel für die Programmentwicklung. Als Spezifikationssprachen dienen sie zur Formulierung von Aufgabenbeschreibungen im allgemeinen oder sind für bestimmte Anwendungsgebiete speziell zugeschnitten. Der Entwurf und die Implementierung solcher Sprachen durch Übersetzer und deren Herstellung durch Generatoren sind die zentralen Themen dieser Veranstaltung.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Teilnehmer sind in der Lage <ul style="list-style-type: none"> <li>• grundlegende Kalküle zur präzisen Beschreibung von Spracheigenschaften anzuwenden,</li> <li>• grundlegende Methoden zur Implementierung von Sprachen einzusetzen.</li> <li>• generierende Werkzeuge zur Sprachimplementierung auszuwählen und zu benutzen.</li> </ul>	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Programmiersprachen und Übersetzer: <ul style="list-style-type: none"> <li>• Vorlesung</li> <li>• Diskussion</li> <li>• Lesen</li> <li>• “Check your Knowledge”, Überprüfung des Lernstands</li> <li>• Übungen</li> <li>• Rechner-Übungen</li> <li>• Sprachimplementierungs-Projekt “SetLan”</li> <li>• Hausaufgaben</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Dr. Peter Pfahler
<b>Lernmaterialien, Literaturangaben</b>
Programmiersprachen und Übersetzer:

- Vorlesungsfolien
- Elektronischer Seminarapparat (koaLA)
- Ebooks
- Handbücher zu den Übersetzer-Werkzeugen

**Sonstige Hinweise**

keine

### 3.27 Pflichtmodul: Programmierung

Modulname	Programmierung / Programming
Workload	240 h
Leistungspunkte	8 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Programmierung : 1</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Programmierung: Vorlesung ( 60h / 150h / DE / WS / 500 ) Programmierung: Übung ( 30h / 0h / DE / WS / 40 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Programmierung: Es sind keine Vorkenntnisse erforderlich	
<b>Inhalte</b>	
<p>Programmierung: Softwareentwicklung ist ein zentrales Arbeitsgebiet der Informatik. Software-Entwickler müssen Aufgaben analysieren und modellieren, Software-Strukturen entwerfen und diese in einer Programmiersprache implementieren können. Dieser Modul vermittelt einführende und wissenschaftlich fundierte Kenntnisse und Fähigkeiten in der Programmierung. Zusammen mit den Modulen Modellierung, Datenbanksysteme, und Softwaretechnik werden damit die wissenschaftlichen Grundlagen für das Arbeitsgebiet Software-Entwicklung gelegt und praktisch eingeübt. Dieses Modul soll die Teilnehmer befähigen,</p> <ul style="list-style-type: none"> <li>• eine für die Software-Entwicklung relevante Programmiersprache anzuwenden (zur Zeit Python, in geringerem Umfang auch Java)</li> <li>• Grundbegriffe der objektorientierten Programmiermethodik einzusetzen,</li> <li>• Algorithmen in Programmen zu implementieren.</li> </ul> <p>Im Informatikstudium bildet dieses Modul zusammen mit den Pflichtmodulen Modellierung, Datenbanksysteme und Softwaretechnik den Kern der Grundausbildung in Gebiet Softwaretechnik.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Die Studierenden lernen ...</p> <p>Faktenwissen, unter anderem - die wesentlichen Konstrukte einer Programmiersprache (derzeit Python, in geringem Umfang auch Java), - die Grundkonzepte von Komposition und Abstraktion in der Programmierung zu verstehen</p> <p>methodisches Wissen - die gelernten Sprachkonstrukte sinnvoll und mit Verständnis anzuwenden, - Software zu testen sowie Fehlerursachen zu finden und zu beseitigen, - objektorientierte Grundkonzepte zu verstehen und anzuwenden, - Software aus objektorientierten Bibliotheken wiederzuverwenden</p>	

Transferkompetenz - praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben zu übertragen normativ-bewertenden Kompetenzen - den Aufwand und die Durchführbarkeit von Programmieraufgabe zu beurteilen
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Lernmotivation</li> </ul>
<b>Methodische Umsetzung</b>
Programmierung: Sprachkonstrukte und Programmieretechniken werden an typischen Beispielen eingeführt und erläutert und anschließend in den Übungen praktisch erprobt. Objektorientierte Methoden und Abstraktion werden überwiegend an der Benutzung von Bibliotheken erklärt. In Übungsstunden in Kleingruppen werden praktische Programmieraufgaben unter Anleitung an Rechnern bearbeitet.
<b>Prüfungsleistung (Dauer)</b>
Klausur (120 - 180 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 8 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Stefan Böttcher
<b>Lernmaterialien, Literaturangaben</b>
Programmierung: keine

<b>Sonstige Hinweise</b>
keine

### 3.28 Pflichtmodul: Rechnerarchitektur

Modulname	Rechnerarchitektur / Computer Architecture
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Rechnerarchitektur : 3</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Rechnerarchitektur: Vorlesung ( 30h / 90h / DE / WS / 300 ) Rechnerarchitektur: Übung ( 30h / 0h / DE / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Rechnerarchitektur: Kenntnisse aus der Lehrveranstaltung Digitaltechnik sind hilfreich.	
<b>Inhalte</b>	
Rechnerarchitektur: Die Vorlesung gibt eine Einführung in den Aufbau und Entwurf moderner Rechner-Systeme. Insbesondere wird vermittelt, wie durch ein effizientes Zusammenspiel von Hardware und Software kostengünstige und leistungsstarke Rechner entwickelt werden können. Die vorgestellten Techniken und Methoden werden in den Übungen an Beispielen vertieft.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage, <ul style="list-style-type: none"> <li>• den Aufbau eines modernen Rechners sowie das Zusammenspiel von Hardware und Software zu beschreiben,</li> <li>• die zugrunde liegenden allgemeinen Entwurfsprinzipien und -strategien zu erklären und anzuwenden,</li> <li>• Rechner-Systeme im Hinblick auf Leistung und Kosten zu analysieren und zu bewerten, sowie</li> <li>• selbstständig einfache Assemblerprogramme zu schreiben.</li> </ul>	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>	
<b>Methodische Umsetzung</b>	
Rechnerarchitektur: <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>	

<ul style="list-style-type: none"> <li>• Rechnerübungen zur Assemblerprogrammierung</li> </ul>
<b>Prüfungsleistung (Dauer)</b>
Klausur (60 - 90 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 5 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Marco Platzner
<b>Lernmaterialien, Literaturangaben</b>
Rechnerarchitektur: <ul style="list-style-type: none"> <li>• Vorlesungsfolien und Übungsblätter</li> <li>• D. A. Patterson, J. L. Hennessy: Computer Organization &amp; Design –The Hardware / Software Interface (3rd Edition); Morgan Kaufmann, 2007; ISBN: 978-0-12-370606-5, ISBN-10: 0-12-370606-8</li> <li>• Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien auf der Webseite und in den Vorlesungsfolien</li> </ul>
<b>Sonstige Hinweise</b>
keine



### 3.29 Wahlpflichtmodul: Rechnernetze

Modulname	Rechnernetze / Computer Networks
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Rechnernetze : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Rechnernetze: Vorlesung ( 45h / 105h / DE / WS / 100 ) Rechnernetze: Übung ( 30h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Rechnernetze: Vorlesung Systemsoftware und systemnahe Programmierung oder vergleichbar.	
<b>Inhalte</b>	
<p>Rechnernetze: Die Vorlesung Rechnernetze behandelt konzeptionelle und technologische Grundlagen von Rechnernetzen/Internet; thematisch werden dabei die Ebenen 1–4 des ISO/OSI-Modells abgedeckt. Zusätzlich werden Ansätze und Werkzeuge zur quantitativen Untersuchung von Kommunikationsprotokollen behandelt. Die Vorlesung wird durch eine Tafelübung begleitet.</p> <ul style="list-style-type: none"> <li>• Die Veranstaltung lässt sich sehr gut mit der Veranstaltung Verteilte Systeme ergänzen.</li> <li>• In einigen Semestern (wenn sowohl Rechnernetze als auch Verteilte Systeme angeboten werden) findet die Veranstaltung halbsemestrig statt; in der zweiten Semesterhälfte die Veranstaltung Verteilte Systeme.</li> </ul>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Absolventen der Lehrveranstaltung</p> <ul style="list-style-type: none"> <li>• können die wesentlichen Aufgaben bei Konstruktion und Bau eines Rechnernetzes benennen und wesentliche Architekturansätze beschreiben;</li> <li>• können unterschiedliche Lösungen für ein Problem aufzählen, deren Vor- und Nachteile herausfinden und sich, gemäß der Anforderungen, für eine Lösung entscheiden;</li> <li>• Schwachstellen existierender Lösungen identifizieren und neue Kommunikationsprotokolle entwickeln und deren Leistungsfähigkeit bewerten.</li> </ul> <p>übung der zugehörigen Lehrveranstaltung</p>	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Rechnernetze: Folienbasierte Vorlesung mit Tafelanschrieb, durch Übung begleitet. Übungen dabei sowohl konzeptionell/analytisch als auch mit praktischen Aufgaben.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Rechnernetze: Folien, Standardlehrbücher (insbes. Tanenbaum, Rechnernetze), Übungsblätter.
<b>Sonstige Hinweise</b>
keine

### 3.30 Pflichtmodul: Schlüsselqualifikation

Modulname	Schlüsselqualifikation
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Mentoring : beliebig</li> <li>• Proseminar : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Mentoring: Treffen in Kleingruppen ( 15h / 15 h / DE / WS oder SS / 20 ) Proseminar: Präsenzzeit Seminar ( 15h / 105 h / DE / WS oder SS / 15 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
Für das Proseminar können alle Proseminare aus dem Angebot des Bachelorstudiengangs Informatik gewählt werden.	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Mentoring: keine Proseminar: Je nach gewähltem Thema.	
<b>Inhalte</b>	
<p>Mentoring: Im Mentoring werden Studierende einzelnen Lehrenden und deren Mitarbeiterinnen bzw. Mitarbeitern in Mentorengruppen (15 – 20 Studierende) zugeordnet. Es finden während des gesamten Bachelorstudiums je nach Bedarf etwa zweimal im Semester Treffen statt. Ziel ist es, durch Beratung – individuell oder in Kleingruppen – Probleme des Studiums und des Fa-ches zu bearbeiten. Dabei sollen Engagement, Motivation und Selbstständigkeit als Aspekte von Selbstkompetenz gestärkt werden. Das Mentoring zielt auf Vermeidung unnötig langer Studiendauern und auf Reduktion der Abbrecherquote.</p> <p>Proseminar: Im Proseminar soll beispielhaft die Einarbeitung in ein wissenschaftliches Thema erlernt und abstraktes Denken gestärkt werden. Die Inhalte sollen schriftlich und mündlich präsentiert werden. Dazu soll Basiswissen in Bezug auf Literaturrecherche, Rhetorik und aktuelle Präsentationstechniken sowie in Bezug auf Kritikfähigkeit und Feedbackmethoden erworben und angewendet werden.</p>	
<b>Lernergebnisse / Fachkompetenzen</b>	
Im Proseminar werden wesentliche Techniken des Erwerbs und der Präsentation wissenschaftlicher Ergebnisse und Erkenntnisse vermittelt. Im Mentoring erhalten Studierenden wichtige Kenntnisse zur erfolgreichen Organisation des Studiums.	

<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Haltung und Einstellung</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Medienkompetenz</li> <li>• Motivationale und volitionale Fähigkeiten</li> <li>• Schreib- und Lesekompetenz (wissenschaftlich)</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Mentoring: Es finden während des gesamten Bachelorstudiums je nach Bedarf etwa zweimal im Semester Treffen statt, in Kleingruppen oder individuell.</p> <p>Proseminar: Referate mit schriftlicher Ausarbeitung und Vortrag.</p>
<b>Prüfungsleistung (Dauer)</b>
<p>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.</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 10 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>Mentoring: keine</p> <p>Proseminar: Je nach gewähltem Thema.</p>

<b>Sonstige Hinweise</b>
keine

### 3.31 Pflichtmodul: Software Engineering

Modulname	Software Engineering / Software Engineering
Workload	150 h
Leistungspunkte	5 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Praktikum: Software Engineering : 2</li> <li>• Software Engineering : 2</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Praktikum: Software Engineering: ( 0h / 30h / DE / SS / 3 ) Software Engineering: Vorlesung ( 30h / 75h / DE / WS / 300 ) Software Engineering: Übung ( 15h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Praktikum: Software Engineering: Programmierung, Modellierung Software Engineering: Programmierung, Modellierung	
<b>Inhalte</b>	
<p>Praktikum: Software Engineering: Begleitend zur Vorlesung Software Engineering werden an einem durchgängigen Beispiel ausgehend von der Spezifikation von Anforderungen sowohl die Modellierung von Softwaresystemen als auch der Übergang von den Modellen zur Implementierung sowie zum modellbasierten Testen der Softwaresysteme bearbeitet.</p> <p>Software Engineering: In der Vorlesung werden die Grundlagen der systematischen und ingenieurmäßigen Softwareentwicklung vermittelt. Im Fokus steht dabei die modellbasierte Softwareentwicklung. Die Vorlesung führt in wesentliche Vorgehensmodelle für die Softwareentwicklung ein, sowohl klassische als auch agile. Es werden Methoden für die Softwareentwicklung und -qualitätssicherung vermittelt, die innerhalb der Vorgehensmodelle zum Einsatz kommen. Außerdem werden Modellierungssprachen und Softwarewerkzeuge vorgestellt, mit denen die statischen und dynamischen Aspekte von Softwaresystemen beschrieben werden können. Insbesondere wird die objektorientierte Modellierungssprache UML (Unified Modeling Language) eingeführt, die unterschiedliche Diagrammsprachen wie Klassendiagramme, Komponentendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme und Zustandsdiagramme vereint. Modellierungswerkzeuge werden exemplarisch eingesetzt. Die Vorlesung wird abgerundet durch eine durchgängige Entwicklungsmethode von der Anforderungsspezifikation über den Architektur- und Softwareentwurf bis hin zur Implementierung und dem Testen der Software. Hierbei wird vor allem auf die Aspekte der systematischen Ableitung und Verfeinerung von Modellen, der Transformation von Modellen in Programmcode (Codegenerierung) sowie des modellbasierten Testens eingegangen. Es werden methodische Hinweise zur Erstellung der Ergeb-</p>	

<p>nisartefakte (u.a. Richtlinien, Architekturstile und Entwurfsmuster) und zur Prüfung ihrer Qualität sowie zum Einsatz der Modellierungssprachen im Softwareentwicklungsprozess gegeben. Darüber hinaus werden Techniken zur Definition und domänenspezifischen Anpassung von Modellierungssprachen (Metamodellierung, UML-Profile sowie Beispiele konkreter domänenspezifischer Sprachen (DSLs) wie SysML oder BPMN) betrachtet. Die Vorlesung wird durch Übungen begleitet, in der die Vorlesungsinhalte aufgegriffen, vertieft und an beispielhaften Entwicklungsaufgaben selbst angewendet werden.</p>
<p><b>Lernergebnisse / Fachkompetenzen</b></p>
<p>Die Studierenden sollen in der Lage sein, für ein gegebenes Problem schrittweise eine Softwarelösung zu entwickeln. Hierzu sollen sie ein modellbasiertes Vorgehen einsetzen können, wobei sie für die einzelnen Entwicklungsschritte unterschiedliche Diagrammart der UML (Unified Modeling Language) verwenden. Zur Überprüfung der Qualität der entwickelten Softwarelösung sollen sie in der Lage sein, Techniken des modellbasierten Testens einzusetzen. Sie verstehen die Beziehungen und Übergänge zwischen verschiedenen Entwicklungsphasen eines Vorgehensmodells. Sie beherrschen verschiedene Diagrammsprachen der UML zur Modellierung der unterschiedlichen Aspekte einer Softwarelösung und können die Qualität von Zwischenergebnissen bewerten. Außerdem haben sie ein grundlegendes Verständnis der Techniken zur Entwicklung und Spezialisierung von Modellierungssprachen für spezielle Situationen und Domänen. Durch den Einsatz des Gelernten am durchgängigen Beispiel der Praktikumsaufgabe verstehen die Studierenden die Bedeutung der verschiedenen Phasen einer Softwareentwicklung und sind in der Lage, diese durchgängig an einem konkreten Softwaresystem einzusetzen.</p>
<p><b>Nichtkognitive Kompetenzen</b></p>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Haltung und Einstellung</li> <li>• Kooperationskompetenz</li> <li>• Lernkompetenz</li> <li>• Lernmotivation</li> <li>• Selbststeuerungskompetenz</li> </ul>
<p><b>Methodische Umsetzung</b></p>
<p>Praktikum: Software Engineering; Softwareentwicklungsaufgaben (Modellieren, Implementieren, Testen) in kleinen Teams.</p> <p>Software Engineering: In der Vorlesung werden die Grundlagen, Begrifflichkeiten, Sprachen und Methoden des Software Engineering vermittelt, die dann in den begleitenden Übungen vertieft und in dem begleitenden Praktikumsanteil von den Studierenden an einem durchgängigen Beispiel selbst erprobt werden.</p>
<p><b>Prüfungsleistung (Dauer)</b></p>
<p>Klausur (60 - 90 Minuten) Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</p>
<p><b>Modulteilprüfungen</b></p>
<p>keine</p>
<p><b>Studienleistung / qualifizierte Teilnahme</b></p>

<p>Studienleistung: schriftliche Übungsaufgaben  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.</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 5 Credits gewichtet.</p>
<p><b>Modul wird in folgenden Studiengängen verwendet</b></p>
<p>–</p>
<p><b>Modulbeauftragte/r</b></p>
<p>Prof. Dr. Gregor Engels</p>
<p><b>Lernmaterialien, Literaturangaben</b></p>
<p>Praktikum: Software Engineering: Aufgabenstellung, Material der begleitenden Vorlesung Software Engineering, eigenständige Recherche zu weiterführender Literatur (in der Bibliothek, im Internet)  Software Engineering: Folien, Tafelanschrieb, evtl. Vorlesungsaufzeichnung, Übungen, Praktikumsaufgabe (siehe Praktikum: Software Engineering)</p>
<p><b>Sonstige Hinweise</b></p>
<p>keine</p>



### 3.32 Wahlpflichtmodul: Softwaremodellierung mit Formalen Methoden

Modulname	Softwaremodellierung mit Formalen Methoden / Softwaremodelling with Formal Methods
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Softwaremodellierung mit formalen Methoden : 6</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Softwaremodellierung mit formalen Methoden: Vorlesung ( 45h / 105h / DE / SS / 90 ) Softwaremodellierung mit formalen Methoden: Übung ( 30h / 0h / DE / SS / 90 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Softwaremodellierung mit formalen Methoden: Kurs Modellierung	
<b>Inhalte</b>	
Softwaremodellierung mit formalen Methoden: Formale Methoden sind Sprachen zur Modellierung/Spezifikation von Systemen. Ein Modell eines (Soft- oder Hardware) Systems beschreibt auf einer gewissen Abstraktionsebene die Funktionalität des Systems. Im Gegensatz zu (den meisten) Programmiersprachen besitzen formale Methoden eine genau festgelegte Semantik, d.h. eine mathematische Beschreibung der Bedeutung einer Spezifikation. Diese Festlegung der Semantik erlaubt es, das Systemmodell bereits vor der eigentlichen Implementierung formal zu analysieren und mögliche Fehler frühzeitig zu finden. In der Vorlesung werden verschiedene formale Methoden eingeführt, die für unterschiedliche Systemarten geeignet sind. Für jede dieser formalen Methoden werden Semantik und Analysetechniken vorgestellt und Modellierungsbeispiele zur Illustration des Einsatzbereiches besprochen. Am Anfang der Vorlesung wird es vorrangig um die Modellierung von Parallelität und Kommunikation gehen. Hier werden Petrinetze und die Prozessalgebra CCS vorgestellt. Danach werden Sprachen zur Beschreibung von zeitlichen Aspekten (Timed Automata) und zustandsbasierte Formalismen zur Spezifikationen von Daten und Operationen (Z und Object-Z) erläutert.	
<b>Lernergebnisse / Fachkompetenzen</b>	

Studierende sind in der Lage, Software- wie Hardwaresysteme formal zu modellieren. Sie können entscheiden, welche Formalismen für die Modellierung am geeignetsten sind. Studierenden können Sicherheitseigenschaften ihrer Modelle analysieren und dafür Werkzeuge einsetzen. Sie besitzen die Fähigkeit die Semantik von neuen Formalismen zu definieren und existierenden Analyseverfahren anzupassen.
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Softwaremodellierung mit formalen Methoden: Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Heike Wehrheim
<b>Lernmaterialien, Literaturangaben</b>
Softwaremodellierung mit formalen Methoden: Ernst-Rüdiger Olderog, Henning Dierks: Real-time Systems (für Abschnitt Timed Automata) Vorlesungsfolien, Übungsaufgaben, evtl. Skript

<b>Sonstige Hinweise</b>
keine

### 3.33 Pflichtmodul: Softwaretechnikpraktikum

Modulname	Softwaretechnikpraktikum / Software Engineering Project
Workload	240 h
Leistungspunkte	8 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Softwaretechnikpraktikum : 3</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Softwaretechnikpraktikum: Vorlesung ( 15h / 150h / DE / WS / 150 ) Softwaretechnikpraktikum: Übung ( 30h / 0h / DE / WS / 10 ) Softwaretechnikpraktikum: ( 45h / 0 h / DE / WS / 150 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Softwaretechnikpraktikum: <ul style="list-style-type: none"> <li>• Objektorientierte Programmierung mit Java</li> <li>• Modellbasierter Softwareentwurf mit UML</li> </ul>	
<b>Inhalte</b>	
Softwaretechnikpraktikum: Das Softwaretechnikpraktikum ist eine praxisorientierte Lehrveranstaltung inklusive Vorlesungen zum Thema Projektmanagement. Eine komplexe Softwareentwicklungsaufgabe wird im Team von ca. zehn Studierenden unter Verwendung von UML und Java bearbeitet. Schwerpunkt des Praktikums ist die Vermittlung von Erfahrungen mit der gruppenbasierten Softwareentwicklung unter Benutzung marktüblicher Werkzeuge, Methoden und Prozesse.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Die Studierenden sollen Sprachen und Werkzeuge im Softwareentwicklungsprozess einsetzen sowie den organisatorischen Ablauf eines Softwareprojekts von der Anforderungsdefinition bis zur Abgabe kennen lernen. Darüber hinaus sollen die Studierenden den praktischen Nutzen von planerisch durchdachten Projekten, sowie die Probleme gruppenorientierter Softwareentwicklung und erste Ansätze zu ihrer Bewältigung kennen lernen.	
<b>Nichtkognitive Kompetenzen</b>	
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Gruppenarbeit</li> <li>• Kooperationskompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>	

<b>Methodische Umsetzung</b>
Softwaretechnikpraktikum: Durchführung eines Projekts mit regelmäßiger Abgabe von Arbeitsergebnissen (Modulteilprüfungen), protokollierten Gruppensitzungen und einer Abschlusspräsentation.
<b>Prüfungsleistung (Dauer)</b>
keine Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Modulteilprüfungen</b>
Klausur (120-180 Minuten, 30% der Modulnote) und Softwareprojekte mit Dokumentation (70% der Modulnote). Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.
<b>Studienleistung / qualifizierte Teilnahme</b>
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.
<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 4 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. Eric Bodden
<b>Lernmaterialien, Literaturangaben</b>
Softwaretechnikpraktikum: <ul style="list-style-type: none"> <li>• Balzert, H. (2008). Lehrbuch der Softwaretechnik. Bd. 3: Softwaremanagement. Hrsg. von C. Ebert. 2. Aufl. Lehrbücher der Informatik. Heidelberg: Spektrum. ISBN: 978-3-8274-1161-7.</li> <li>• Balzert, H. (2011). Lehrbuch der Softwaretechnik. Bd. 2: Entwurf, Implementierung, Installation und Betrieb. 3. Aufl. Lehrbücher der Informatik. Heidelberg: Spektrum. ISBN: 978-3-8274-1706-0.</li> <li>• Sommerville, I. (2012). Software Engineering. 9. Aufl. Always Learning. München: Pearson. ISBN: 978-3-86894-099-2.</li> </ul>
<b>Sonstige Hinweise</b>

Im Softwaretechnikpraktikum 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.

### 3.34 Pflichtmodul: Stochastik für Informatiker

Modulname	Stochastik für Informatiker / Stochastic for Computer Science
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Stochastik für Informatiker : 3</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Stochastik für Informatiker: Vorlesung ( 45h / 105h / DE / WS / 200 ) Stochastik für Informatiker: Übung ( 30h / 0h / DE / WS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Erfolgreicher Abschluss der Module Analysis für Informatiker und Lineare Algebra für Informatiker	
<b>Empfohlene Kenntnisse</b>	
Stochastik für Informatiker:	
<b>Inhalte</b>	
<p>Stochastik für Informatiker: Deskriptive Statistik und Datenanalyse, Klassische Wahrscheinlichkeitsmodelle, Axiomatik, Standardverteilungen (u.a. Binomial), Satz von Bayes und Anwendungen, Bsp. für nicht-dis-krete Ws.räume, Zufallsgrößen und ihre Momente, Quantile, Gesetze der großen Zahlen, Zent-raler Grenzwertsatz, Schätzen (inkl.-Konfidenzintervalle) und Testen, Simulation und Zufalls-zahlen, Markovketten, mehrdimensionale Wahrscheinlichkeitsverteilungen Deskriptive Statistik und Datenanalyse</p> <ul style="list-style-type: none"> <li>• planen statistische Erhebungen (Befragung, Beobachtung oder Experiment), führen sie durch und werten sie aus</li> <li>• lesen und erstellen grafische Darstellungen für uni- und bivariate Daten (z.B. Kreuzta-belle) und bewerten deren Eignung für die jeweilige Fragestellung</li> <li>• bestimmen und verwenden uni- und bivariate Kennwerte (z.B. Mittelwerte, Streumaße, Korrelationen, Indexwerte) und interpretieren sie angemessen Zufallsmodellierung</li> <li>• modellieren mehrstufige Zufallsversuche durch endliche Ergebnismengen und nutzen geeignete Darstellungen (Baumdiagramm, Mehrfeldertafel)</li> <li>• rechnen und argumentieren mit Wahrscheinlichkeiten, bedingten Wahrscheinlichkeiten, Erwartungswerten und stochastischer Unabhängigkeit</li> <li>• erläutern inhaltlich das Bernoullische Gesetz der großen Zahlen und den zentralen Grenzwertsatz und deren Konsequenzen</li> <li>• verwenden diskrete und kontinuierliche Verteilungen und ihre Eigenschaften zur Mo-dellierung Stochastische Anwendungen</li> <li>• kennen Beispiele für die Anwendung von Stochastik in verschiedenen Wissenschaften (ökonomie, Physik, ...)</li> <li>• schätzen in Zufallssituationen Parameter aus Daten</li> </ul>	

<ul style="list-style-type: none"> <li>• führen Hypothesentests durch und reflektieren deren zentralen Schritte und bestimmen Konfidenzintervalle</li> <li>• erläutern Unterschiede zwischen Bayes-Statistik und klassischen Testverfahren Neue Medien</li> <li>• verwenden Tabellenkalkulation und statistische Software zur Darstellung und explorativen Analyse von Daten</li> <li>• simulieren Zufallsversuche computergestützt</li> </ul>
<b>Lernergebnisse / Fachkompetenzen</b>
<ul style="list-style-type: none"> <li>• Kenntnis der Bedeutung der Stochastik in Gesellschaft und Wissenschaft.</li> <li>• Sicherer Umgang mit den Begriffen der Stochastik in Wort und Schrift.</li> <li>• Verständnis des mathematischen Sachverhaltes und den damit verbundenen Denkweisen.</li> <li>• Verständnis der Beweise. Befähigung zur Lösung von Übungsaufgaben zur Stochastik. Fähigkeit des Erkennens von Verbindungen innerhalb der Stochastik beziehungsweise zwischen der Stochastik und anderen Bereichen der Mathematik.</li> <li>• Durchführung von einfachen statistischen Analysen. Befähigung zum Umgang mit einem Software-Paket zur Stochastik.</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Stochastik für Informatiker:
<b>Prüfungsleistung (Dauer)</b>
<p>Klausur (90 - 120 Minuten)</p> <p>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</p> <p>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. Jürgen Klüners
<b>Lernmaterialien, Literaturangaben</b>
Stochastik für Informatiker: keine
<b>Sonstige Hinweise</b>
keine

### 3.35 Pflichtmodul: Studium Generale

Modulname	Studium Generale / General Studies
Workload	210 h
Leistungspunkte	7 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Studium Generale : beliebig</li> <li>• Studium Generale : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Studium Generale: Vorlesung ( 45h / 135h / DE / WS / 30 ) Studium Generale: Übung ( 30h / 0h / DE / WS / 30 ) 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. Studium Generale: Abhängig von den gewählten Veranstaltungen.	
<b>Inhalte</b>	
Studium Generale: Abhängig von den gewählten Veranstaltungen. 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.
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 7 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. Studium Generale: Abhängig von den gewählten Veranstaltungen.
<b>Sonstige Hinweise</b>
Eine beliebige Kombination von Veranstaltungen außerhalb der Informatik und im Umfang von maximal 7 LP muss gewählt werden. Nebenfach und Studium Generale haben einen Umfang von insgesamt 25 LP. Die angegebene Verteilung der LP auf Lehrveranstaltungen ist nur exemplarisch.

### 3.36 Pflichtmodul: Systemsoftware und systemnahe Programmierung

Modulname	Systemsoftware und systemnahe Programmierung / System software and system-level programming
Workload	270 h
Leistungspunkte	9 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Praktikum: Systemsoftware und systemnahe Programmierung : 4</li> <li>• Systemsoftware und systemnahe Programmierung : 4</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Praktikum: Systemsoftware und systemnahe Programmierung : ( 0h / 30h / DE / SS / 3 ) Systemsoftware und systemnahe Programmierung : Vorlesung ( 60h / 150h / EN / SS / 200 ) Systemsoftware und systemnahe Programmierung : Übung ( 30h / 0h / EN / SS / 25 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
keine	
<b>Empfohlene Kenntnisse</b>	
Praktikum: Systemsoftware und systemnahe Programmierung : Grundlagen der Programmierung.  Systemsoftware und systemnahe Programmierung : Es ist dringend zu empfehlen, die Vorlesungen Programmierung und Modellierung erfolgreich abgeschlossen zu haben. Ebenso sollten Grundlagen der Rechnerarchitektur bekannt sein.	
<b>Inhalte</b>	
Praktikum: Systemsoftware und systemnahe Programmierung : Begleitend zur Vorlesung Systemsoftware und systemnahe Programmierung werden in diesem Programmierpraktikum Techniken der systemnahen Programmierung praktisch erprobt und eingeübt. Studierende werden in konkreten Projekten das Problem analysieren, geeignete Programmier Techniken auswählen, praktisch realisieren und eine quantitative Leistungsbewertung durchführen.  Systemsoftware und systemnahe Programmierung : Einführung in grundlegende Probleme, Aufgaben, Herausforderungen und Herangehensweisen für systemnahe Software (z.B. Betriebssysteme, Protokollstacks). Es wird ein konzeptioneller Zugang gewählt (anstelle eines beispielorientierten Ansatzes); besonderer Wert wird auf praktisch orientierte Programmierübungen in kleinen Projekten gelegt, die den selbständigen Umgang mit der Materie vertiefen.	
<b>Lernergebnisse / Fachkompetenzen</b>	
Studierende können Aufgabenstellungen der Systemsoftware identifizieren, unterschiedliche Ansätze zu Problemlösungen benennen, klassifizieren und unterscheiden, deren Vor- und Nachteile evaluieren und	

<p>für ein Problem eine geeignete Lösung auswählen. Sie sind in der Lage, diese Verfahren in eigenen Anwendungen gezielt zum Einsatz zu bringen (bspw. Semaphoren zur Koordination nebenläufiger Aktivitäten).</p> <p>Studierende können ggf. neue Lösungen konstruieren (bspw. Scheduling-Strategie) und deren Leistungsfähigkeit systematisch durch Einsatz geeigneter (mathematischer oder informatischer) Werkzeuge analysieren, deren Eignung evaluieren und mit Alternativen kontrastieren.</p>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Gruppenarbeit</li> <li>• Lernkompetenz</li> <li>• Selbststeuerungskompetenz</li> </ul>
<b>Methodische Umsetzung</b>
<p>Praktikum: Systemsoftware und systemnahe Programmierung : Programmieraufgaben in kleineren Teams.</p> <p>Systemsoftware und systemnahe Programmierung : Die Vorlesung ist überwiegend folienorientiert, mit begleitendem Tafelinsatz und Aufgaben für die Studierenden während der Vorlesung. Sie wird sowohl durch Tafelübung als auch durch Kleingruppentutorien begleitet. Studierende haben in den Kleingruppen Gelegenheit, Aufgaben in der Gruppe zu bearbeiten und Übungsblätter durch Tutoren benoten zu lassen.</p>
<b>Prüfungsleistung (Dauer)</b>
<p>Klausur (120 - 180 Minuten)</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>
<p>Studienleistung: schriftliche Übungsaufgaben</p> <p>Qualifizierte Teilnahme: Praktikumsarbeit mit anschließendem Gespräch</p> <p>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 9 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–

<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Praktikum: Systemsoftware und systemnahe Programmierung : Aufgabenstellung, man pages, eigenständige Recherche zu unterstützender Kommunikation. Systemsoftware und systemnahe Programmierung : Standardlehrbücher (z.B. Stallings, Betriebssysteme); Foliensatz der VL; Übungsblätter.
<b>Sonstige Hinweise</b>
keine

### 3.37 Wahlpflichtmodul: Verteilte Algorithmen und Datenstrukturen

Modulname	Verteilte Algorithmen und Datenstrukturen / Distributed Algorithms and Data Structures
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Verteilte Algorithmen und Datenstrukturen : beliebig</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Verteilte Algorithmen und Datenstrukturen: Vorlesung ( 45h / 105h / DE / SS / 60 ) Verteilte Algorithmen und Datenstrukturen: Übung ( 30h / 0h / DE / SS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Verteilte Algorithmen und Datenstrukturen: Datenstrukturen und Algorithmen	
<b>Inhalte</b>	
Verteilte Algorithmen und Datenstrukturen: Die Vorlesung gibt eine Einführung in die Grundlagen der verteilten Algorithmen und Datenstrukturen. Einen Schwerpunkt bilden dabei hochskalierbare Datenstrukturen, die auch für sehr große und dynamische verteilte Systeme anwendbar sind. Nach einer Einführung in die Netzwerktheorie werden zunächst grundlegende Designprinzipien für verteilte Algorithmen und Datenstrukturen vorgestellt wie z.B. das Konzept der selbst-stabilisierenden Systeme. Danach folgt eine kurze Einführung in verteilte Programmierung, damit die in der Vorlesung vorgestellten Datenstrukturen auch von den Studenten implementiert werden können. Anschließend werden zunächst prozessorientierte Datenstrukturen und dann informationsorientierte Datenstrukturen vorgestellt.	
<b>Lernergebnisse / Fachkompetenzen</b>	
<ul style="list-style-type: none"> <li>• Kenntnis ausgewählter verteilter Algorithmen und Datenstrukturen</li> <li>• Kenntnis wesentlicher Konzepte im Bereich verteilter Algorithmen und Datenstrukturen</li> <li>• Fähigkeit, selbstständig adäquate Techniken und Verfahren im Bereich der verteilten Algorithmen und Datenstrukturen zu entwickeln</li> <li>• Fähigkeit, algorithmische Probleme gemäß ihrer Lösbarkeit und Komplexität einzuschätzen</li> </ul>	

<ul style="list-style-type: none"> <li>• Fähigkeit, grundlegende verteilte Datenstrukturen zu implementieren</li> </ul>
<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>
Verteilte Algorithmen und Datenstrukturen: Vorlesung mit Übungen, Hausaufgaben und Softwareprojekt
<b>Prüfungsleistung (Dauer)</b>
<p>Klausur (90 - 120 Minuten)</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>
<p>Studienleistung: schriftliche Übungsaufgaben</p> <p>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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Christian Scheideler
<b>Lernmaterialien, Literaturangaben</b>
Verteilte Algorithmen und Datenstrukturen: Skript
<b>Sonstige Hinweise</b>
keine



### 3.38 Wahlpflichtmodul: Verteilte Systeme

Modulname	Verteilte Systeme / Distributed Systems
Workload	180 h
Leistungspunkte	6 LP
Studiensemester	<ul style="list-style-type: none"> <li>• Verteilte Systeme : 5</li> </ul>
<b>Lehrveranstaltungen: Lehrform ( Kontaktzeit / Selbststudium / Sprache / Termin / Gruppengröße )</b>	
Verteilte Systeme: Vorlesung ( 45h / 105h / DE / WS / 100 ) Verteilte Systeme: Übung ( 30h / 0h / DE / WS / 30 )	
<b>Wahlmöglichkeiten innerhalb des Moduls</b>	
keine	
<b>Teilnahmevoraussetzungen</b>	
Die Module Programmierung, Programmiersprachen, Software Engineering, Datenbanksysteme, Modellierung, Datenstrukturen und Algorithmen, Digitaltechnik, Analysis für Informatiker und Lineare Algebra für Informatiker müssen bestanden sein. Bei Studierenden des Nebenfachs Mathematik werden dabei die Module „Analysis 1“ und „Lineare Algebra 1“ statt der Module „Lineare Algebra für Informatiker“ und „Analysis für Informatiker“ berücksichtigt.	
<b>Empfohlene Kenntnisse</b>	
Verteilte Systeme: Vorlesung Systemsoftware und systemnahe Programmierung. Grundlegendes Verständnis von Algorithmen.	
<b>Inhalte</b>	
<p>Verteilte Systeme: Diese Veranstaltung behandelt architekturelle, konzeptionelle und pragmatische Fragestellungen beim Entwurf, Einsatz und Betrieb von verteilten Systemen in der Informatik – Systeme, bei denen Daten oder Kontrollfunktionen nicht mehr an einem Ort konzentriert sind sondern die sich aus unabhängigen IT-Systemen zusammensetzen. Dabei wird der Systemaspekt betont; grundlegende algorithmische Fragestellungen werden ebenfalls behandelt. Zusätzlich werden Fragen der Leistungsbeurteilung und Verlässlichkeit behandelt.</p> <p>Bemerkungen:</p> <ul style="list-style-type: none"> <li>• Die Veranstaltung lässt sich sehr gut mit der Veranstaltung Rechnernetze ergänzen.</li> <li>• In der Regel findet die Veranstaltung halbjährlich in der zweiten Semesterhälfte statt; in der ersten Semesterhälfte die Veranstaltung Rechnernetze.</li> </ul>	
<b>Lernergebnisse / Fachkompetenzen</b>	
<p>Teilnehmer sind in der Lage,</p> <ul style="list-style-type: none"> <li>• verteilte Systeme zur Erhöhung von Leistungsfähigkeit oder Fehlertoleranz zum Einsatz zu bringen und geeignet zu dimensionieren;</li> <li>• sie können geeignete Systemansätze (Client-Server, P2P, ...) benennen und situationsgerecht auswählen und diese Auswahl architekturell begründen;</li> </ul>	

<ul style="list-style-type: none"> <li>• sie haben algorithmische Problemstellungen für verteilte Systeme verstanden, können aus einer allgemeinen Problembeschreibung die zu lösenden algorithmische Aufgabe isolieren und eine begründete Wahl treffen.</li> </ul>
<b>Nichtkognitive Kompetenzen</b>
<ul style="list-style-type: none"> <li>• Einsatz und Engagement</li> <li>• Lernkompetenz</li> </ul>
<b>Methodische Umsetzung</b>
Verteilte Systeme: Folienbasierte Vorlesung mit Tafelanschrieb, durch Übung begleitet. Übungen dabei sowohl konzeptionell/analytisch als auch mit praktischen Aufgaben.
<b>Prüfungsleistung (Dauer)</b>
Klausur (90 - 120 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 12 Credits gewichtet.
<b>Modul wird in folgenden Studiengängen verwendet</b>
–
<b>Modulbeauftragte/r</b>
Prof. Dr. rer. nat. Holger Karl
<b>Lernmaterialien, Literaturangaben</b>
Verteilte Systeme: Folien, Standardlehrbücher (insbes. Colouris, Distributed Systems Concepts and Design; Tanenbaum, Verteilte Systeme), Übungsblätter.
<b>Sonstige Hinweise</b>

keine



# Anhang A

## Überblickstabellen

### A.1 Studienrichtungen und Module

	Erster Studienabschnitt (S. 11)	Informatikgebiet Algorithmen und Komplexität (S. 13)	Informatikgebiet Computer Systeme (S. 14)	Informatikgebiet Daten und Wissen (S. 15)	Informatikgebiet Softwaretechnik (S. 16)	Informatikgebiet Vertiefung (S. 17)
Analysis für Informatiker (S. 20)	X	-	-	-	-	-
Angriffssicherer Softwareentwurf (S. 23)	-	-	-	-	X	X
Bachelor-Abschlussarbeit (S. 25)	-	-	-	-	-	-
Berechenbarkeit und Komplexität (S. 28)	X	-	-	-	-	-
Betriebssysteme (S. 30)	-	-	X	-	-	X
Computer Graphics Rendering (S. 32)	-	-	-	X	-	X
Data Mining (S. 35)	-	-	-	X	-	X
Databases and Information Systems (S. 37)	-	-	-	X	-	X
Datenbanksysteme (S. 40)	X	-	-	-	-	-
Datenstrukturen und Algorithmen (S. 43)	X	-	-	-	-	-
Digitaltechnik (S. 46)	X	-	-	-	-	-
Einführung in Kryptographie (S. 48)	-	X	-	-	-	X
Eingebettete Systeme (S. 50)	-	-	X	-	-	X
Gestaltung von Nutzungsschnittstellen (S. 53)	X	-	-	-	-	-
Grundlagen Wissensbasierter Systeme (S. 55)	-	-	-	X	-	X

Modellbasierte Softwareentwicklung (S. 72)	-	-	-	-	X	X
Modellierung (S. 74)	X	-	-	-	-	-
Parallelität und Kommunikation (S. 76)	-	X	-	-	-	X
Programmiersprachen (S. 78)	X	-	-	-	-	-
Programmiersprachen und Übersetzer (S. 80)	-	-	-	-	X	X
Programmierung (S. 83)	X	-	-	-	-	-
Rechnerarchitektur (S. 86)	X	-	-	-	-	-
Rechnernetze (S. 88)	-	-	X	-	-	X
Schlüsselqualifikation (S. 90)	X	-	-	-	-	-
Software Engineering (S. 93)	X	-	-	-	-	-
Softwaremodellierung mit Formalen Methoden (S. 96)	-	-	-	-	X	X
Softwaretechnikpraktikum (S. 99)	X	-	-	-	-	-
Stochastik für Informatiker (S. 102)	X	-	-	-	-	-
Studium Generale (S. 105)	-	-	-	-	-	-
Systemsoftware und systemnahe Programmierung (S. 107)	X	-	-	-	-	-
Verteilte Algorithmen und Datenstrukturen (S. 110)	-	X	-	-	-	X
Verteilte Systeme (S. 112)	-	-	X	-	-	X

## A.2 Module und Lehrveranstaltungen



---

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

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

---

**ISSN 2199-2819**