UNIVERSITÄT PADERBORN

FAKULTÄT FÜR ELEKTROTECHNIK, INFORMATIK UND MATHEMATIK

INSTITUT FÜR INFORMATIK

MODULHANDBUCH FÜR DEN BACHELORSTUDIENGANG INFORMATIK V4 (IBA V4)

STAND: 30. SEPTEMBER 2026

Inhaltsverzeichnis

1 Präambel und Hinweise **3**

2 Erster Studienabschnitt **4**
2.1 Softwaretechnik 4
2.2 Algorithmen und Komplexität 22
2.3 Computersysteme 32
2.4 Mathematik 45

3 Zweiter Studienabschnitt **55**
3.1 Algorithmen und Komplexität 55
3.2 Computersysteme 83
3.3 Daten und Wissen 101
3.4 Softwaretechnik 121
3.5 Vertiefung 142

4 Weiteres **143**

5 Module im Wintersemester **151**

6 Module im Sommersemester **152**

7 Modules in English **153**

1 Präambel und Hinweise

Aus technischen Gründen wurde die Präambel des Modulhandbuches ausgelagert. Sie ist unter Prüfungsangelegenheiten auf den Seiten zum Studium des Instituts für Informatik zu finden. Wir bitten um Beachtung dieser Präambel. Bei Fragen zu dieser Präambel wenden Sie sich bitte an das Studienbüro Informatik.

Bitte beachten Sie auch, dass

1. in diesem Modulhandbuch alle laut Prüfungsordnung vorgesehenen Module aufgelistet werden, auch wenn sie in dem entsprechenden Semester nicht angeboten werden.
2. dieses Modulhandbuch den Datenbestand zum Zeitpunkt der Erstellung beinhaltet. Alle Angaben sind ohne Gewähr.

2 Erster Studienabschnitt

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.

2.1 Softwaretechnik

<table>
<tr><th colspan="5">Programmierung</th></tr>
<tr><td colspan="5">Programming</td></tr>
<tr><td colspan="2">Modulnummer:
M.079.1110</td><td>Workload (h):
240</td><td>Leistungspunkte:
8</td><td>Turnus:
module.frequency.de</td></tr>
<tr><td colspan="2"></td><td>Studiensemester:
1. Semester</td><td>Dauer (in Sem.):
1</td><td>Sprache:
de</td></tr>
<tr><td>1</td><td colspan="4">Modulstruktur
<table>
<tr><th></th><th>Lehrveranstaltung</th><th>Lehr-form</th><th>Kontakt-zeit (h)</th><th>Selbst-studium (h)</th><th>Status (P/WP)</th><th>Gruppen-größe (TN)</th></tr>
<tr><td>a)</td><td>L.079.05100
Programmierung</td><td>V4 Ü2</td><td>90</td><td>150</td><td>P</td><td>500/40</td></tr>
</table></td></tr>
<tr><td>2</td><td colspan="4">Wahlmöglichkeiten innerhalb des Moduls:
keine</td></tr>
<tr><td>3</td><td colspan="4">Teilnahmevoraussetzungen:
keine</td></tr>
</table>

4	**Inhalte:** *Inhalte der Lehrveranstaltung 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 Programmierung 2, Programmiersprachen, Softwareengineering und dem Softwaretechnikpraktikum werden damit die wissenschaftlichen Grundlagen für das Arbeitsgebiet Software-Entwicklung gelegt und praktisch eingeübt. Dieses Modul soll die Teilnehmer befähigen, • eine für die Software-Entwicklung relevante Programmiersprache anzuwenden (zur Zeit Python, in geringerem Umfang auch Java) • Grundbegriffe der objektorientierten Programmiermethodik einzusetzen, • Algorithmen in Programmen zu implementieren. Im Informatikstudium bildet dieses Modul zusammen mit den Pflichtmodulen Modellierung, Datenbanksysteme und Softwaretechnik den Kern der Grundausbildung in Gebiet Softwaretechnik. Dieses Modul umfasst die folgenden Inhalte: 1. Grundbegriffe zu Programmen und ihrer Ausführung 2. Klassen, Objekte, Datentypen 3. Programm- und Datenstrukturen 4. Objektorientierte Abstraktion 5. Objektorientierte Bibliotheken
5	**Lernergebnisse und Kompetenzen:** Die Studierenden lernen • *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 • *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 • *Transferkompetenz:* praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben zu übertragen • *normativ-bewertenden Kompetenzen:* den Aufwand und die Durchführbarkeit von Programmieraufgabe zu beurteilen **Nichtkognitive Kompetenzen:** • Gruppenarbeit • Kooperationskompetenz • Lernmotivation

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur</td><td>120-180 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Stefan Böttcher</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Programmierung:
Methodische Umsetzung
Sprachkonstrukte und Programmiertechniken 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 Übungenstunden in Kleingruppen werden praktische Programmieraufgaben unter Anleitung an Rechnern bearbeitet.</td></tr>
</table>

Programmiersprachen			
Programming Languages			
Modulnummer: M.079.1111	**Workload (h):** 120	**Leistungspunkte:** 4	**Turnus:** module.frequency.de
	Studiensemester: 1. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05101 Programmiersprachen	V2 Ü1	45	75	P	500/40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Programmiersprachen:
Empfohlene Vorkenntnisse
Die Veranstaltung Programmierung bzw. Kenntnisse im Umgang mit (mindestens einer) Programmiersprache.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Programmiersprachen:
In der Veranstaltung “Programmierung” haben Sie bereits eine grundlegende Einführung in die Programmierung mit *einer* sogenannten *imperativen* Programmiersprache genossen, die Berechnungen durch Sequenzen einzelner Befehle beschreiben. Die Veranstaltung “Programmiersprachen” baut hierauf auf und vermittelt Einblicke in *andere* wichtige sogenannte Programmierparadigmen, speziell in die funktionale Programmierung und Logikprogrammierung.
Inhaltsübersicht:

- Grundlegende Konzepte der Funktionalen Programmierung
- Seiteneffekte, Programmieren ohne Seiteneffekte
- Rekursion und Aggregation
- Grundlegende Konzepte der Logikprogrammierung
- Auswertung von Ausdrücken und Regeln, Unifizierung
- Logikprogrammierung in Prolog
- Eingeschränkte aber effiziente Auswertung in Datalog

5 **Lernergebnisse und Kompetenzen:**

- Die Studierenden verstehen die grundlegenden syntaktischen und semantischen Konzepte von Programmiersprachen.
- Sie können Programme und die darin verwendeten Sprachkonstrukte in angemessen präziser Weise erläutern.
- Sie sind in der Lage, einfache Grammatiken, Typspezifikationen und funktionale Programme zu entwickeln.
- Sie besitzen die Fähigkeit neue Programmier- und Anwendungssprachen selbstständig zu erlernen.
- Sie haben praktisch mit einer Reihe verschiedener Programmiersprachen gearbeitet.
- Sie haben Einblick gewonnen in die Probleme des Entwurfs neuer Programmier- und Anwendungssprachen.
- Sie können typische Eigenschaften von objektorientierten und funktionalen Sprachen erklären.

Nichtkognitive Kompetenzen:

- Lernmotivation
- Einsatz und Engagement
- Gruppenarbeit
- Kooperationskompetenz
- Schreib- und Lesekompetenz (wissenschaftlich)
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	60-90 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10	**Gewichtung für Gesamtnote:** Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4)
12	**Modulbeauftragte/r:** Prof. Dr. Eric Bodden, Prof. Dr. Stefan Böttcher
13	**Sonstige Hinweise:**

Datenbanksysteme

Database Systems

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.1114	150	5	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	2. Semester	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05203 Datenbanksysteme	V2 Ü2	60	90	P	400/40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Datenbanksysteme:
Empfohlene Vorkenntnisse
Kenntnisse in der Programmierung werden in dem Umfang vorausgesetzt, wie sie in den Veranstaltungen Programmierung und Programmiersprachen gelehrt werden. Elementare Kenntnisse der Logik der Modellierung aus der Vorlesung Modellierung werden ebenfalls vorausgesetzt.

4	**Inhalte:** *Inhalte der Lehrveranstaltung 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. • Relationales Datenmodell, relationale Algebra und relationale Kalküle • SQL (Datendefinitonssprache, Datenmanipulationssprache und Anfragesprache) • Eingebettetes SQL • Sichten, Zugriffskontrolle und View-Update-Problematik • Anfrageoptimierung • Datenintegrität • Funktionale Abhängigkeiten und Datenbankschemaentwurf • Transaktionen (Synchronisation und Recovery) • NoSQL-Datenbanken
5	**Lernergebnisse und Kompetenzen:** Studierende lernen • *Faktenwissen:* Theorie und Konzepte relationaler Anfragesprachen kennen; Konzepte des Datenbankentwurfs kennen; Konzepte der Synchronisation und Recovery von Transaktionen kennen • *Vermittlung von methodischem Wissen in Kleingruppen-Präsenz-Übungen:* komplexe Anfragen an relationale Datenbanken korrekt formulieren; ein Datenbankschema möglichst redundanzfrei entwerfen • *in praktischen Übungen am Rechner:* eigene SQL-Anfragen an existierende relationale Datenbanken stellen; Programme zu schreiben, die Datenbestände aus Datenbanken lesen oder verändern; eigene Datenbanken zu definieren und aufzubauen • *Vermittlung von Transferkompetenz:* die erworbenen Kompetenzen und Fertigkeiten auf andere Datenquellen oder andere Datenbanksysteme übertragen; Umgang mit Zugriffsrechten • *Vermittlung von normativ-bewertenden Kompetenzen:* die Eignung und Grenzen des relationalen Datenmodells bewerten und einschätzen; den Programmieraufwand für Datenbankanfragen und Datenbankprogrammierung einschätzen; die Folgen einer Datenbankschema-Änderung erkennen und abschätzen; die Risiken eines schlecht entworfenen Datenbankschemas bewerten; den Aufwand und Nutzen von Synchronisation und Recovery abschätzen **Nichtkognitive Kompetenzen:** • Gruppenarbeit • Lernkompetenz • Lernmotivation

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	60-90 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Dr. Rita Hartel, Prof. Dr. Stefan Böttcher

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Datenbanksysteme:
Methodische Umsetzung
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.
Lernmaterialien, Literaturangaben

- Lehrbuch: Kemper, Eickler: Datenbanksysteme , Oldenbourg-Verlag, neueste Ausgabe.
- Lehrbuch: Garcia-Molina, Ullman, Widom: Database Systems: The Complete Book, Prentice Hall, neueste Ausgabe.

Software Engineering			
Software Engineering			
Modulnummer: M.079.1112	**Workload (h):** 150	**Leistungspunkte:** 5	**Turnus:** module.frequency.de
	Studiensemester: 2. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05202 Software Engineering	V2 Ü1	45	75	P	300/30
b)	L.079.05206 Praktikum: Software Engineering	P	0	30	P	3

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Software Engineering:
Empfohlene Vorkenntnisse
Programmierung, Modellierung

Teilnahmevoraussetzungen der Lehrveranstaltung Praktikum: Software Engineering:
Empfohlene Vorkenntnisse
Programmierung, Modellierung

4	**Inhalte:** *Inhalte der Lehrveranstaltung Software Engineering:* In der Vorlesung werden die Grundlagen der systematischen und ingenieursmäßigen 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 Ergebnisartefakte (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 denen die Vorlesungsinhalte aufgegriffen, vertieft und an beispielhaften Entwicklungsaufgaben selbst angewendet werden. Schwerpunkte der Veranstaltung sind: • Vorgehensmodelle (klassische, agile) • UML (Unified Modelling Language): Klassendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme, Zustandsdiagramme • Modellbasiertes Vorgehensmodell • durchgängige Softwareentwicklungsmethode von der Anforderungspezifikation über Modellierung bis zur Implementierung und dem Test der Software • Modellbasiertes Testen • Domänenspezifische Sprachen (Metamodellierung, UML Profile, SysML, BPMN) *Inhalte der Lehrveranstaltung Praktikum: Software Engineering:* Begleitend zur Vorlesung Software Engineering vertiefen die Studierenden anhand von praktischen Übungsaufgaben ihre Kenntnisse in der Programmiersprache Java, welche häufig zur systematischen Entwicklung größerer Softwaresysteme verwendet wird. Ein Schwerpunkt liegt dabei auf der technischen Umsetzung von Objektorientierten Konzepten und UML-Spezifikationen (insbesondere Klassendiagrammen) in ausführbare Softwareapplikationen. Die Studierenden erlernen anhand praktischer Beispiele die Zusammenhänge zwischen Entwurf und Implementierung von Softwareanwendungen. Außerdem erwerben sie die Kompetenz, bekannte Programmierkonzepte, welche anhand der Sprache Python erlernt wurden, auf eine weitere Programmiersprache und neue Anwendungsgebiete zu transferieren.

5 **Lernergebnisse und Kompetenzen:**

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 Diagrammarten 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 verschieden 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.
Nichtkognitive Kompetenzen:

- Einsatz und Engagement
- Haltung und Einstellung
- Kooperationskompetenz
- Lernkompetenz
- Lernmotivation
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a) - b)	Klausur	60-90 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL
b)	Praktikumsarbeit mit anschließendem Gespräch		QT

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10	**Gewichtung für Gesamtnote:** Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4)
12	**Modulbeauftragte/r:** Dr. Stefan Sauer
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Software Engineering:* **Methodische Umsetzung** 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. **Lernmaterialien, Literaturangaben** Folien, Tafelanschrieb, evtl. Vorlesungsaufzeichnung, Übungen, Praktikumsaufgabe (siehe Praktikum: Software Engineering) *Hinweise der Lehrveranstaltung Praktikum: Software Engineering:* **Methodische Umsetzung** Softwareentwicklungsaufgaben (Modellieren, Implementieren, Testen) in kleinen Teams. **Lernmaterialien, Literaturangaben** Aufgabenstellung, Material der begleitenden Vorlesung Software Engineering, eigenständige Recherche zu weiterführender Literatur (in der Bibliothek, im Internet)

Softwaretechnikpraktikum			
Software Engineering Project			
Modulnummer: M.079.1113	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 3. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehr-form**	**Kontakt-zeit (h)**	**Selbst-studium (h)**	**Status (P/WP)**	**Gruppen-größe (TN)**
a)	L.079.05300 Softwaretechnikpraktikum	V1 Ü2	45	195	P	150/10

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Softwaretechnikpraktikum:
Empfohlene Vorkenntnisse
Die Module *Programmierung* und *Software Engineering* sollten erfolgreich abgeschlossen sein. Kenntnisse in objektorientierter Programmierung mit Java sowie dem modellbasierten Softwareentwurf mit UML sind empfohlen.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Softwaretechnikpraktikum:
Das Softwaretechnikpraktikum ist ein praxisorientiertes Modul inklusive Vorlesungen zum Thema Projektmanagement. Eine komplexe Softwareentwicklungsaufgabe wird im Team von ca. zehn Studierenden unter Verwendung aktueller Entwicklungsmethoden und der Programmiersprache Java bearbeitet. Schwerpunkt des Praktikums ist die Vermittlung von Erfahrungen mit der gruppenbasierten Softwareentwicklung unter Benutzung marktüblicher Werkzeuge, Methoden und Prozesse.
Die Lehrinhalte umfassen:

- Planung des Projektmanagements
- Erstellung von Projektplänen, Aufwandsschätzungen, Anforderungsspezifikationen und Entwurfsdokumentationen
- Umsetzung der Softwareentwicklungsaufgabe unter Einsatz etablierter Softwareentwicklungsmethoden, wie z.B. Scrum

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- Herausforderungen der arbeitsteiligen Entwicklung einer Softwarelösung benennen und erklären
- aktuelle Softwareentwicklungsmethodiken benennen, beschreiben und praktisch anwenden
- verschiedene Verfahren zur Softwarequalitätssicherung benennen, unterscheiden, miteinander vergleichen und in Teilen auch praktisch anwenden
- die Softwarearchitektur und die Codequalität von Drittanbieterkomponenten erkennen und beurteilen
- ein laufendes Softwaresystem bestehend aus mehreren Komponenten mittels verschiedener Technologien arbeitsteilig entwickeln und dokumentieren.

6 **Prüfungsleistung:**

□Modulabschlussprüfung (MAP) □Modulprüfung (MP) ⊠Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Software mit Dokumentation, Projektpräsentation, schriftliche Hausarbeit	i.d.R. bis zu 2500 Zeilen Code, bis zu 4 Seiten, 30 min, bis zu 10 Seiten	45%, 25%, 30%

Die Prüfungsleistung besteht aus einem Softwareprojekt mit in der Regel bis zu 2500 Zeilen Code sowie Dokumentation im Umfang von in der Regel bis zu 4 Seiten pro Gruppenmitglied, einer Projektpräsentation von ca. 30 Minuten sowie einer schriftlichen Hausarbeit im Umfang von in der Regel bis zu 10 Seiten. Das Softwareprojekt bildet 45% der Modulnote, die Projektpräsentation 25% und die schriftliche Hausarbeit 30%.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Praktikumsarbeit		SL

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulteilprüfungen bestanden sind.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der Hälfte der Anzahl seiner Credits gewichtet (Faktor 0,5).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12	**Modulbeauftragte/r:** Prof. Dr. Eric Bodden
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Softwaretechnikpraktikum:* **Methodische Umsetzung** Durchführung eines Projekts mit regelmäßiger Abgabe von Arbeitsergebnissen (Modulteilprüfungen), protokollierten Gruppensitzungen und einer Abschlusspräsentation. **Lernmaterialien, Literaturangaben** • 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. • 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. • Sommerville, I. (2012). Software Engineering. 9. Aufl. Always Learning. München: Pearson. ISBN: 978-3-86894-099-2.

Gestaltung von Nutzungsschnittstellen			
Designing User Interfaces			
Modulnummer: M.079.1115	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 4. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05400 Gestaltung von Nutzungsschnittstellen	V4 Z1	75	105	P	300

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Gestaltung von Nutzungsschnittstellen:

- Software muss benutzbar sein, um ihren Zweck erfüllen zu können: schlechte Bedienbarkeit bedeutet in der Regel Frustration, Verschwendung von Ressourcen wie Zeit, Aufmerksamkeit und Geld, sowie Ausgrenzung “nicht bedachter” Nutzergruppen.
- In dieser Veranstaltung erlernen wir die Planung, das Design und die Implementierung von benutzbarer Software, insbesondere Nutzungsschnittstellen.
- Wir gehen dabei auf Benutzbarkeit (Usability) ein im Sinne von Erlernbarkeit, Beeinträchtigungsfreiheit (Accessibility), Inklusion, Ästhetik, Gestaltprinzipien, Design Patterns und Gamification und arbeiten auf klassischen sowie mobilen Umgebungen.
- Wir bauen dabei auf physiologischen und psychologischen Grundlagen auf und gehen auch auf (un)ethischen Umgang mit diesen in der Softwareentwicklung ein.

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden sind in der Lage, empirisch und empathisch Anforderungen ihrer Nutzer*innen zu erheben, konstruktiv gestaltend umzusetzen und zu evaluieren. Weiterhin findet eine Begriffsbildung statt, die im Anschluss den Einstieg in weiterführende Veranstaltungen sowie später auch Forschung im Bereich der Mensch-Maschine-Interaktion, auch mit der Schnittstelle Empirische Softwaretechnik und IT-Sicherheit, erleichtert.

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur</td><td>90-120 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Übungsaufgaben</td><td></td><td>SL</td></tr></table></td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Yasemin Acar</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Gestaltung von Nutzungsschnittstellen:
Methodische Umsetzung:
Die Vorlesung ist mit interaktiven Lernquizzes gestaltet; Übungsgruppen dienen der praktischen Anwendung von Gelerntem sowie Klärung von Fragen bei der Bearbeitung der Hausübungen. Die Hausübungen werden von Tutor*innen bewertet und sind, möglicherweise gemeinsam mit den Quizzes, Grundlage für die Studienleistung.
Lernmaterialien, Literaturangaben:
Vorlesungsaufzeichnungen, Vorlesungsfolien (möglicherweise auf Englisch) und Übungsblätter.</td></tr>
</table>

2.2 Algorithmen und Komplexität

Modellierung			
Modelling			
Modulnummer: M.079.1210	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 1. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05102 Modellierung	V4 Ü2	90	150	P	500/40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung 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.

- Grundlagen für Kalküle: Mengen, Terme, Relationen, Beweismethoden
- Modellierung mit Graphen: Wegprobleme, Erreichbarkeit, Bäume, Färbbarkeit
- Logik: Aussagenlogik, Folgerung, Resolution, Prädikatenlogik
- Modellierung von Abläufen: reguläre Ausdrücke, endliche Automaten
- Grammatiken: reguläre und kontextfreie Grammatiken

5 Lernergebnisse und Kompetenzen:

Studierende kennen wesentliche Techniken zur Modellierung informatischer Probleme. Sie können für ein gegebenes Problem eine geeignete Modellierungstechnik auswählen und das Problem mit dieses Technik beschreiben. Sie können grundlegende Techniken erweitern und verfeinern, um so neuartige Probleme zu modellieren.
Nichtkognitive Kompetenzen:

- Lernkompetenz
- Motivationale und volitionale Fähigkeiten

6 Prüfungsleistung:

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	120-180 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 Voraussetzungen für die Teilnahme an Prüfungen:

Bestehen der Studienleistung

9 Voraussetzungen für die Vergabe von Credits:

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 Gewichtung für Gesamtnote:

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).

11 Verwendung des Moduls in anderen Studiengängen:

Bachelorstudiengang Informatik v4 (IBA v4)

12 Modulbeauftragte/r:

Prof. Dr. Christian Scheideler, Prof. Dr.-Ing. Anni-Yasmin Turhan

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Modellierung:* **Methodische Umsetzung** 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. **Lernmaterialien, Literaturangaben** • Uwe Kastens, Hans Kleine Büning, Modellierung • Angelika Steger, Diskrete Strukturen • Foliensatz der Vorlesung; Übungsblätter

Datenstrukturen und Algorithmen			
Data Structures and Algorithms			
Modulnummer: M.079.1211	**Workload (h):** 270	**Leistungspunkte:** 9	**Turnus:** module.frequency.de
	Studiensemester: 2. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05201 Datenstrukturen und Algorithmen	V4 Ü2 Z1	105	135	P	400/25
b)	L.079.05205 Praktikum: Datenstrukturen und Algorithmen	P	0	30	P	3

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Datenstrukturen und Algorithmen:
Empfohlene Vorkenntnisse:
Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse u. a. mit mathematischen Methoden zu erlernen

Teilnahmevoraussetzungen der Lehrveranstaltung Praktikum: Datenstrukturen und Algorithmen:
Empfohlene Vorkenntnisse:

- Grundlagen der Programmierung

4	**Inhalte:** *Inhalte der Lehrveranstaltung 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. • Einführung: Rechenmodelle, Effizienzmaße, Beispiele • Sortierverfahren: Quicksort, Heapsort, Mergesort • Datenstrukturen: Verkette Listen, Bäume, Graphen, Dynamische Suchbäumen, Hashing, Prioritätswarteschlangen • Entwurfs- und Analyseverfahren: Rekursion und das Mastertheorem, Teile-und-Herrsche, Dynamische Programmierung, Backtracking, Branch & Bound, Greedy Algorithmen • Graphenalgorithmen: Kürzeste Wege, Minimale Spannbäume *Inhalte der Lehrveranstaltung 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, geeignete Programmiertechniken auswählen, praktisch realisieren und eine quantitative Leistungsbewertung durchführen. • Sortieralgorithmen • Einfache Graph-Algorithmen • Algorithmen für Wege-Probleme • Algorithmen zur Berechnung minimaler Spannbäume • Wörterbücher • Hashing

5 **Lernergebnisse und Kompetenzen:**

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
- selbstständig und kreative Algorithmen und Datenstrukturen entwickeln (wie gestalte ich den kreativen Prozess vom algorithmischen Problem zum effizienten Algorithmus?)
- mathematischer Methoden zum Korrektheitsbeweis und zur Effizienzanalyse einzusetzen.
- die Wechselwirkung zwischen Algorithmus und Datenstruktur an wesentlichen Beispielen erläutern.
- die Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten einschätzen

Nichtkognitive Kompetenzen:

- Einsatz und Engagement
- Gruppenarbeit
- Haltung und Einstellung
- Lernkompetenz
- Motivationale und volitionale Fähigkeiten
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a) - b)	Klausur	120-180 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL
b)	Praktikumsarbeit mit anschließendem Gespräch		QT

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung und die qualifizierte Teilnahme konkret zu erbringen sind.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung und erfolgreiche Absolvierung des Praktikums Datenstrukturen und Algorithmen

9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4)
12	**Modulbeauftragte/r:** Prof. Dr. Sevag Gharibian
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Datenstrukturen und Algorithmen:* **Methodische Umsetzung:** • Vorlesung mit Beamer und Tafelanschrieb • Übungen in Kleingruppen • erwartete Aktivitäten der Studierenden: aktive Mitarbeit bei Präsenzübungen, Hausaufgaben • Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt • In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt. **Lernmaterialien, Literaturangaben:** • Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter *Hinweise der Lehrveranstaltung Praktikum: Datenstrukturen und Algorithmen:* **Methodische Umsetzung:** • Programmieraufgaben in kleineren Teams **Lernmaterialien, Literaturangaben:** • Aufgabenstellungen, man pages, eigenständige Recherche zu weiterführender Literatur

Berechenbarkeit und Komplexität			
Computability and Complexity			
Modulnummer: M.079.1212	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 3. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05301 Berechenbarkeit und Komplexität	V3 Ü2	75	105	P	200/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Berechenbarkeit und Komplexität:
Erfolgreicher Abschluss der Module Modellierung und Datenstrukturen und Algorithmen

4 **Inhalte:**

Inhalte der Lehrveranstaltung Berechenbarkeit und Komplexität:
Einführung in grundlegende Methoden und Techniken zur Charakterisierung der Schwierigkeit von Berechnungsproblemen. Als formales Rechenmodel 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. Zeitkomplexiät, Klassen P und NP, NP-Vollständigkeit, polynomielle Reduktionen) definiert und erläutert.

- Einführung: Sprachen, Rechenmodelle, Grammatiken, Simulationen
- Berechenbarkeit: Entscheidbare, unentscheidbare Sprachen, Diagonalisierung, Halteproblem, Reduktionen, Beispiele
- Zeitkomplexität: Laufzeiten, Klassen P und NP, polynomielle Reduktionen, NP-Vollständigkeit, SAT, Satz von Cook-Levin, Beispiele
- Approximationsalgorithmen

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
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.
Nichtkognitive Kompetenzen:
• Haltung und Einstellung
• Selbststeuerungskompetenz</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur</td><td>90-120 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Johannes Blömer</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Berechenbarkeit und Komplexität:* **Methodische Umsetzung** 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. **Lernmaterialien, Literaturangaben** • Michael Sipser: Introduction to the Theory of Computation • Uwe Schöning: Theoretische Informatik - kurz gefasst • Foliensatz der Vorlesung; Übungsblätter

2.3 Computersysteme

Digitaltechnik

Digital Design

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.1310	150	5	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	2. Semester	1	de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05200 Digitaltechnik	V2 Ü2	60	90	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Digitaltechnik:
Empfohlene Vorkenntnisse
Kenntnisse aus der Lehrveranstaltung Modellierung sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Digitaltechnik:
Die Veranstaltung 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 mit modernen Entwurfswerkzeugen umgesetzt.
Im einzelnen werden die folgenden Themen behandelt:

- Darstellung von Information und Fehlerkorrigierende Codes
- Boolesche Algebra
- Gatter und Schaltnetze
- Logikoptimierung (Optimierung zweistufiger Logik nach Quine/McCluskey)
- Automaten und Schaltwerke (festverdrahtet, mikroprogrammierbar)
- Arithmetische Einheiten als Entwurfsbeispiele
- Entwurf auf Register-Transfer-Ebene
- Hardware-Beschreibungssprachen und Entwurf mit VHDL

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden sind in der Lage
• den Entwurfsablauf in der Digitaltechnik von der Spezifikation bis zur technischen Realisierung zu beschreiben,
• die zugrunde liegenden mathematischen Modelle aus der Booleschen Algebra und der Automatentheorie anzuwenden,
• digitale Schaltungsentwürfe im Hinblick auf vorgegebene Entwurfsziele zu analysieren,
• einfache digitale Systeme selbständig zu konzipieren, sowie
• einfache digitale Systeme mit den entsprechenden Entwurfswerkzeugen technisch zu realisieren.
Nichtkognitive Kompetenzen:
• Gruppenarbeit
• Lernkompetenz</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur</td><td>60-90 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Marco Platzner</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Digitaltechnik:* **Methodische Umsetzung** • Vorlesung mit Beamer und Tafel • Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer • Praktische Übungen zum Hardware-Entwurf (Teamarbeit) **Lernmaterialien, Literaturangaben** • Vorlesungsfolien und Übungsblätter • Aufgabenblätter und technische Dokumentation für die Rechnerübungen • J. F. Wakerly, „Digital Design,“ 4th Edition, Upper Saddle River, NJ: Pearson / Prentice Hall, 2007 • Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien im jeweiligen PANDA-Kurs

Rechnerarchitektur			
Computer Architecture			
Modulnummer: M.079.1311	**Workload (h):** 150	**Leistungspunkte:** 5	**Turnus:** module.frequency.de
	Studiensemester: 3. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehr-form**	**Kontakt-zeit (h)**	**Selbst-studium (h)**	**Status (P/WP)**	**Gruppen-größe (TN)**
a)	L.079.05302 Rechnerarchitektur	V2 Ü2	60	90	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Rechnerarchitektur:
Empfohlene Vorkenntnisse
Kenntnisse aus der Lehrveranstaltung Digitaltechnik sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Rechnerarchitektur:
Die Vorlesung gibt eine Einführung in den Aufbau und Entwurf moderner Rechensysteme. 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.

- Grundstrukturen, von Neumann Rechner
- Befehlssätze und Assemblerprogrammierung
- Leistungsbewertung
- Datenpfad und Steuerung
- Pipelining
- Speicherhierarchie, insbesondere Cache-Management und virtueller Speicher
- Ein-/Ausgabe

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,
• den Aufbau eines modernen Rechners sowie das Zusammenspiel von Hardware und Software zu beschreiben,
• die zugrunde liegenden allgemeinen Entwurfsprinzipien und -strategien zu erklären und anzuwenden,
• Rechnersysteme im Hinblick auf Leistung und Kosten zu analysieren und zu bewerten, sowie
• selbständig einfache Assemblerprogramme zu schreiben.
Nichtkognitive Kompetenzen:
• Gruppenarbeit
• Lernkompetenz</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur</td><td>60-90 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Christian Plessl, Prof. Dr. Marco Platzner</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Rechnerarchitektur:* **Methodische Umsetzung** • Vorlesung mit Beamer und Tafelanschrieb • Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer • Rechnerübungen zur Assemblerprogrammierung **Lernmaterialien, Literaturangaben** • Vorlesungsfolien und Übungsblätter • D. A. Patterson, J. L. Hennessy: Computer Organization & Design – The Hardware / Software Interface (3rd Edition); Morgan Kaufmann, 2007; ISBN: 978-0-12-370606-5, ISBN-10: 0-12-370606-8 • Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien auf der Webseite und in den Vorlesungsfolien

Systemsoftware und systemnahe Programmierung			
System software and system-level programming			
Modulnummer: M.079.1312	**Workload (h):** 270	**Leistungspunkte:** 9	**Turnus:** module.frequency.de
	Studiensemester: 4. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05401 Systemsoftware und systemnahe Programmierung	V4 Ü2 Z1	105	135	P	200/25
b)	L.079.05402 Praktikum: Systemsoftware und systemnahe Programmierung	P	0	30	P	3

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Systemsoftware und systemnahe Programmierung:
Empfohlene Vorkenntnisse
Es ist dringend zu empfehlen, die Vorlesungen Programmierung und Modellierung erfolgreich abgeschlossen zu haben. Ebenso sollten Grundlagen der Rechnerarchitektur bekannt sein.

Teilnahmevoraussetzungen der Lehrveranstaltung Praktikum: Systemsoftware und systemnahe Programmierung:
Empfohlene Vorkenntnisse
Grundlagen der Programmierung

4	**Inhalte:** *Inhalte der Lehrveranstaltung 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. • Prozessmodell, Prozesswechsel • Scheduling • Zusammenhang mit grundlegenden Aspekten der Rechnerarchitektur (z. B. Interrupt) • Prozesskoordination (z. B. Semaphore) • Betriebsmittelverwaltung (z. B. Deadlocks, Bankier-Algorithmus) • Speicherverwaltung (virtueller Speicher, Paging, . . .) • Interprozesskommunikation, Client/Server • Grundlagen Rechnernetze, Internet • Systemnahe Programmierung (z. B. Socket-Schnittstelle, Thread-Bibliotheken) *Inhalte der Lehrveranstaltung Praktikum: Systemsoftware und systemnahe Programmierung:* Begleitend zur Vorlesung Systemsoftware und systemnahe Programmierung werden in diesem Programmierpraktikum Techniken der systemnnahen Programmierung praktisch erprobt und eingeübt. Studierende werden in konkreten Projekte das Problem analysieren, geeignete Programmiertechniken auswählen, praktisch realisieren und eine quantitative Leistungsbewertung durchführen. Mögliche Projektthemen sind: • Thread-Programmierung: Nutzung bestehender Bilbliotheken; Realisierung einer User-Level Thread-Bibliothek • Socket-Programmierung: Implementierung einfacher Socket-Kommunikation; insbesondere Leistungsbewertung unterschiedlicher Varianten • Scheduling-Algorithmen implementieren • Automatisierung durch Shell-Skripte
5	**Lernergebnisse und Kompetenzen:** Studierende können Aufgabenstellungen der Systemsoftware identifizieren, unterschiedliche Ansätze zu Problemlösungen benennen, klassifizieren und unterscheiden, deren Vor- und Nachteile evaluieren und 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). 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. Praktische Erfahrung im Umgang mit und Einsatz von systemnaher Werkzeuge und Programmiertechniken. **Nichtkognitive Kompetenzen:** • Gruppenarbeit • Lernkompetenz • Selbststeuerungskompetenz

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a) - b)</td><td>Klausur</td><td>120-180 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr><tr><td>b)</td><td>Praktikumsarbeit</td><td></td><td>QT</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung und die qualifizierte Teilnahme konkret zu erbringen sind.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Dr. Florian Klingler</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Systemsoftware und systemnahe Programmierung:* **Methodische Umsetzung** Die Vorlesung ist überwiegend folienorientiert, mit begleitendem Tafeleinsatz 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. **Lernmaterialien, Literaturangaben** Standardlehrbücher (z. B. Stallings, Betriebssysteme); Foliensatz der Vorlesung; Übungsblätter. *Hinweise der Lehrveranstaltung Praktikum: Systemsoftware und systemnahe Programmierung:* **Methodische Umsetzung** • Programmieraufgaben in kleineren Teams. **Lernmaterialien, Literaturangaben** • Aufgabenstellung, man pages, eigenständige Recherche zu unterstützender Kommunikation.

IT-Sicherheit			
IT Security			
Modulnummer: M.079.1313	**Workload (h):** 150	**Leistungspunkte:** 5	**Turnus:** module.frequency.de
	Studiensemester: 5. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehr-form**	**Kontakt-zeit (h)**	**Selbst-studium (h)**	**Status (P/WP)**	**Gruppen-größe (TN)**
a)	L.079.05500 IT-Sicherheit	V2 Ü2	60	90	P	150/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung IT-Sicherheit:
Empfohlene Vorkenntnisse:
Rechnernetze, Programmierung, Systemsoftware und systemnahe Programmierung

4 **Inhalte:**

Inhalte der Lehrveranstaltung 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 vorgestellt und geeignete Gegenmaßnahmen diskutiert. Hierzu gehört auch die Vorstellung praxisrelevanter kryptographischer Protokolle und Algorithmen sowie deren Sicherheitseigenschaften.
In der Veranstaltung werden folgende Themen behandelt:

- Ausgewählte Sicherheitsprobleme
- Grundlagen der angewandten Kryptographie
- Sicherheitsziele und Sicherheitsmodelle
- Netzwerksicherheit
- Anwendungssicherheit

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Studierende verstehen die wesentlichen Konzepte, Methoden und Mechanismen zum Schutz von Daten und Systemen vor Manipulation und Missbrauch auf einem grundlegenden, praxisorientierten, 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.
Nichtkognitive Kompetenzen:
• Einsatz und Engagement
• Lernkompetenz
• Lernmotivation</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur</td><td>90-120 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Juraj Somorovsky</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung IT-Sicherheit:* **Methodische Umsetzung:** • Vorlesung mit Beamer und Tafelanschrieb • Präsenzübungen in kleinen Gruppen mit Übungsblättern zu den theoretischen Grundlagen, Präsentation der Lösungen durch Übungsteilnehmer • Praktische Übungen zur IT Sicherheit **Lernmaterialien, Literaturangaben:** • Vorlesungsfolien und Übungsblätter • Sicherheit und Kryptographie im Internet, Jörg Schwenk • Computer Security, William Stallings und Lawrie Brown

2.4 Mathematik

Analysis für Informatiker

Calculus for Computer Science

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.105.9710	240	8	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	1. Semester	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.105.96100 Analysis für Informatiker	V4 Ü2	90	150	P	450/40

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4	**Inhalte:** *Inhalte der Lehrveranstaltung Analysis für Informatiker:* Die in diesen Veranstaltungen vermittelten Kenntnisse werden als Verfahren bzw. Faktenwissen im Informatikstudium gebraucht; bzw. die mathematisch-methodische Denkweise (Definition, Satz, Beweis), die hier eingeübt wird. 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
5	**Lernergebnisse und Kompetenzen:** Die Studierenden • beschreiben den progressiven Aufbau des Zahlensystems (bis hin zu den komplexen Zahlen) und argumentieren mit dem Permanenzprinzip als formaler Leitidee, • verwenden die Begriffe der Konvergenz von Folgen und Reihen sowie der Vollständigkeit der reellen Zahlen formal sicher und erläutern diese Begriffe an tragenden Beispielen, • 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, • definieren den Begriff des Integrals formal und verwenden ihn in mathematischen Zusammenhängen, interpretieren das Integrieren als Flächenmessung und als Mittelwertbildung, • erläutern und begründen den Hauptsatz der Differential- und Integralrechnung, • nutzen Software zur Darstellung und Exploration mathematischer Modellierungen und als heuristisches Werkzeug zur Lösung von Anwendungsproblemen, • kennen und reflektieren Fragen der Umsetzung numerischer Verfahren auf dem Computer (z.B. Komplexität, Genauigkeit). **Nichtkognitive Kompetenzen:** • Lernkompetenz • Selbststeuerungskompetenz

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur</td><td>120-180 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Jürgen Klüners</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Analysis für Informatiker:
Methodische Umsetzung
Die Studierenden
• präsentieren und erklären mathematische Sachverhalte
• denken konzeptionell, analytisch und logisch
• denken und handeln eigenständig
• erarbeiten sich interessengelenkt selbstständig mathematische Einsichten</td></tr>
</table>

Lineare Algebra für Informatiker			
Linear Algebra for Computer Science			
Modulnummer: M.105.9720	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 2. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.105.96200 Lineare Algebra für Informatiker	V4 Ü2	90	150	P	300/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung 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 Approximation) für alle Sparten der Mathematik, für die mathematische Modellbildung und für die mathematischen Anwendungen hat.

1. Grundbegriffe
2. Vektorräume
3. lineare Abbildungen
4. Basis
5. Dimension
6. Matrizen
7. lineare Gleichungssysteme
8. Determinanten
9. Eigenwerte
10. charakteristisches Polynom
11. Normalformenproblem

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden

- 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,
- begreifen lineare Abbildungen von Vektorräumen als strukturverträgliche Abbildungen und erläutern, wie lineare Gleichungssysteme koordinatenfrei durch sie beschrieben werden,
- verstehen den abstrakten Basis- und Dimensionsbegriff und erklären, wie dieser als Verallgemeinerung des naiven Koordinaten- und Dimensionsbegriff verstanden werden kann,
- stellen lineare Abbildungen durch Matrizen dar und begreifen diese als koordinatenabhängige Realisierung,
- verstehen und erläutern, wie sich die (eindeutige) Lösbarkeit solcher Gleichungssysteme charakterisieren lässt; lösen lineare Gleichungssysteme und erklären Lösungsverfahren,
- 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,
- kennen den Begriff des Eigenwerts; verstehen und erklären das Normalformenproblem, kennen Kriterien für Diagonalisierbarkeit

Nichtkognitive Kompetenzen:

- Lernkompetenz
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	120-180 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10	**Gewichtung für Gesamtnote:** Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4)
12	**Modulbeauftragte/r:** Prof. Dr. Jürgen Klüners
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Lineare Algebra für Informatiker:* **Methodische Umsetzung** Die Studierenden • reflektieren eigene Lernerfahrungen • präsentieren und erklären mathematische Sachverhalte • denken konzeptionell, analytisch und logisch • erarbeiten sich interessengeleitet selbständig neue Erkenntnisse • denken und handeln eigenständig

Stochastik für Informatiker			
Stochastic for Computer Science			
Modulnummer: M.105.9730	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 3. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehr-form**	**Kontakt-zeit (h)**	**Selbst-studium (h)**	**Status (P/WP)**	**Gruppen-größe (TN)**
a)	L.105.96300 Stochastik für Informatiker	V3 Ü2	75	105	P	200/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Erfolgreicher Abschluss mindestens eines der Module Analysis für Informatiker oder Lineare Algebra für Informatiker

4	**Inhalte:** Deskriptive Statistik und Datenanalyse, Klassische Wahrscheinlichkeitsmodelle, Axiomatik, Standardverteilungen (u.a. Binomial), Satz von Bayes und Anwendungen, Bsp. für nicht-diskrete Ws.räume, Zufallsgrößen und ihre Momente, Quantile, Unabhängigkeit und mehrdimensionale Wahrscheinlichkeitsverteilungen, Gesetze der großen Zahlen, Zentraler Grenzwertsatz, Schätzen (inkl.-Konfidenzintervalle) und Testen, Simulation und Zufallszahlen, Markovketten, Unabhängigkeit und mehrdimensionale Wahrscheinlichkeitsverteilungen *Inhalte der Lehrveranstaltung Stochastik für Informatiker:* Deskriptive Statistik und Datenanalyse, Klassische Wahrscheinlichkeitsmodelle, Axiomatik, Standardverteilungen (u.a. Binomial), Satz von Bayes und Anwendungen, Bsp. für nicht-diskrete Wahrscheinlichkeitsräume, Zufallsgrößen und ihre Momente, Quantile, Gesetze der großen Zahlen, Zentraler Grenzwertsatz, Schätzen (inkl. Konfidenzintervalle) und Testen, Simulation und Zufallszahlen, Markovketten, mehrdimensionale Wahrscheinlichkeitsverteilungen Deskriptive Statistik und Datenanalyse • planen statistische Erhebungen (Befragung, Beobachtung oder Experiment), führen sie durch und werten sie aus • lesen und erstellen grafische Darstellungen für uni- und bivariate Daten (z.B. Kreuztabelle) und bewerten deren Eignung für die jeweilige Fragestellung • bestimmen und verwenden uni- und bivariate Kennwerte (z.B. Mittelwerte, Streumaße, Korrelationen, Indexwerte) und interpretieren sie angemessen Zufallsmodellierung • modellieren mehrstufige Zufallsversuche durch endliche Ergebnismengen und nutzen geeignete Darstellungen (Baumdiagramm, Mehrfeldertafel) • rechnen und argumentieren mit Wahrscheinlichkeiten, bedingten Wahrscheinlichkeiten, Erwartungswerten und stochastischer Unabhängigkeit • erläutern inhaltlich das Bernoullische Gesetz der großen Zahlen und den zentralen Grenzwertsatz und deren Konsequenzen • verwenden diskrete und kontinuierliche Verteilungen und ihre Eigenschaften zur Modellierung Stochastische Anwendungen • kennen Beispiele für die Anwendung von Stochastik in verschiedenen Wissenschaften (Ökonomie, Physik, . . .) • schätzen in Zufallssituationen Parameter aus Daten • führen Hypothesentests durch und reflektieren deren zentralen Schritte und bestimmen Konfidenzintervalle • erläutern Unterschiede zwischen Bayes-Statistik und klassischen Testverfahren Neue Medien • verwenden Tabellenkalkulation und statistische Software zur Darstellung und explorativen Analyse von Daten • simulieren Zufallsversuche computergestützt

5 **Lernergebnisse und Kompetenzen:**

- Kenntnis der Bedeutung der Stochastik in Gesellschaft und Wissenschaft.
- Sicherer Umgang mit den Begriffen der Stochastik in Wort und Schrift.
- Verständnis des mathematischen Sachverhaltes und den damit verbundenen Denkweisen.
- 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.
- Durchführung von einfachen statistischen Analysen. Befähigung zum Umgang mit einem Software-Paket zur Stochastik.

Nichtkognitive Kompetenzen:

- Lernkompetenz
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	90-120 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Prof. Dr. Martin Kolb

13	**Sonstige Hinweise:**

3 Zweiter Studienabschnitt

Im zweiten Studienabschnitt haben die Studierenden die Gelegenheit, sich in Fächern ihrer Wahl gezielt zu vertiefen. Gleichzeitig wird weiterhin die Breite des Studiums garantiert, indem alle Studierenden mindestens eine weiterführende Veranstaltung aus den vier Gebieten Algorithmen und Komplexität, Computersysteme, Daten und Wissen sowie Softwaretechnik belegen müssen. Weiter gehören zu diesem Studienabschnitt ein Proseminar und die Bachelorarbeit.

3.1 Algorithmen und Komplexität

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.

3D Modeling (BA Inf v4)			
3D Modeling (BA Inf v4)			
Modulnummer: M.079.2232	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05540 3D Modeling	V3 Ü2	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung 3D Modeling:
Die Veranstaltung vermittelt grundlegende Konzepte und Techniken der Computergrafik im Bereich der 3D-Modellierung. Verschiedene wichtige digitale Repräsentationsformen für zwei- und dreidimensionale Objekte (Kurven, Flächen, Volumen) und Welten, ihr jeweiliges mathematisches Fundament und damit jeweils zusammenhängende grundlegende Datenstrukturen und Algorithmen werden besprochen. Betrachtet werden dabei folgende Repräsentationsformen:

- Polygonnetze
- Polyedernetze
- Netze höherer Ordnung
- Subdivision-Techniken
- Bézier-Techniken
- Splines & NURBS
- CAD-BRep
- Implizite Repräsentation
- Point/Splat-basierte Repräsentation
- Radiance Fields

5 Lernergebnisse und Kompetenzen:

Die Studierenden können

- die verschiedenen Repräsentationsformen für dreidimensionale Daten benennen, beschreiben und gegenüberstellen.
- die verschiedenen Repräsentationsformen für dreidimensionale Daten hinsichtlich ihrer relativen Eignung für konkrete Anwendungsfälle bewerten und auswählen.
- mit diesen Repräsentationsformen zusammenhängende grundlegende Datenstrukturen und Algorithmen reproduzieren, implementieren und anwenden.
- die mathematischen Grundlagen von Subdivision-Techniken erklären.
- die mathematischen Grundlagen von Bézier- und Spline-Techniken erklären und die daraus folgenden formellen Eigenschaften diskutieren.

6 Prüfungsleistung:

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8 Voraussetzungen für die Teilnahme an Prüfungen:

Bestehen der Studienleistung

9 Voraussetzungen für die Vergabe von Credits:

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 Gewichtung für Gesamtnote:

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 Verwendung des Moduls in anderen Studiengängen:

Bachelorstudiengang Informatik v4 (IBA v4)

12 Modulbeauftragte/r:

Prof. Dr. Marcel Campen

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung 3D Modeling:* **Methodische Umsetzung** Inhalte werden im Rahmen der Vorlesung unter Nutzung von Präsentationsfolien und Tafelanschrieb vermittelt. In Präsenzübungen sowie in Heimarbeit werden die Inhalte vertieft und durch praktische Übungen ergänzt. **Lernmaterialien, Literaturangaben** • Vorlesungsfolien, Übungsblätter • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

3D Rendering (BA Inf v4)			
3D Rendering (BA CS v4)			
Modulnummer: M.079.2231	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 6	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05641 3D Rendering	V3 Ü2	75	105	P	60/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung 3D Rendering:
Die Veranstaltung vermittelt grundlegende Konzepte, Techniken und Algorithmen der Computergrafik im Bereich der 3D-Bilderzeugung. Mathematische Aspekte aus dem Bereich der linearen Algebra kommen zum Einsatz zur räumlichen Modellierung, physikalische Aspekte aus dem Bereich der geometrischen Optik zur Modellierung von Beleuchtung, Sichtbarkeit und Schatten. Betrachtet werden auch die Umsetzung in Software und die Ausnutzung spezialisierter Hardware.
Die Veranstaltung umfasst folgende Inhalte:

- Geometrische Primitive
- Räumliche Transformationen
- Projektion und Perspektive
- Beleuchtung und Farben
- Sichtbarkeit und Schatten
- Rasterisierung
- Raytracing
- GPUs und APIs
- Shader
- Texture-Mapping
- Bildfilterung

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Nach Abschluss des Moduls können die Studierenden
• die wesentlichen Bestandteile der Rasterisierungspipeline benennen und beschreiben
• die Rasterisierungspipeline in einfachster Form selbst implementieren
• unter Nutzung eines API fremde Implementierungen der Rasterisierungspipeline anwenden
• einfache Raytracing-Verfahren erklären und implementieren
• räumliche Transformations- und Projektionsoperationen auswählen und anwenden
• die Schwierigkeiten der realitätsnahen Beleuchtungssimulation diskutieren
• Methoden zur approximativen Beleuchtungssimulation gegenüberstellen und bewerten
• Vorwärts- und Rückwärts-Verfahren des Renderings differenzieren
• die Grenzen der Echtzeitcomputergrafik diskutieren</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 30-45 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Übungsaufgaben, schriftliche Ausarbeitung oder Fortschrittsberichte</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Marcel Campen</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung 3D Rendering:* **Methodische Umsetzung** Inhalte werden im Rahmen der Vorlesung unter Nutzung von Präsentationsfolien und Tafelanschrieb vermittelt. In Präsenzübungen sowie in Heimarbeit werden die Inhalte vertieft und durch praktische Übungen ergänzt. **Lernmaterialien, Literaturangaben** • Vorlesungsfolien, Kompaktskript, Übungsblätter. • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Einführung in Kryptographie (BA Inf v4)

Introduction to Cryptography (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2210	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05503 Einführung in Kryptographie	V3 Ü2	75	105	WP	100/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Kryptographie:
Empfohlene Vorkenntnisse
Datenstrukturen und Algorithmen sowie Berechenbarkeit und Komplexität

4 **Inhalte:**

Inhalte der Lehrveranstaltung 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.

- Aufgaben der Kryptographie
- Symmetrische und asymmetrische Verfahren
- Elementare Sicherheitskonzepte und Kryptanalyse
- Symmetrische Verschlüsselungsverfahren - DES, AES
- Asymmetrische Verschlüsselungsverfahren - RSA, Elgamal
- Schlüsselaustauschverfahren - Diffie-Hellman
- Hashfunktionen und MACs - SHA3
- Digitale Signaturen - RSA, Elgamal, DSA

5 **Lernergebnisse und Kompetenzen:**

Studierende sind in der Lage Sicherheitsanforderungen mittels kryptographischer Aufgaben zu formulieren. Sie kennen die wichtigsten kryptographischen Basistechniken und ihre Einsatzmöglichkeiten. 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.

Nichtkognitive Kompetenzen:

- Lernkompetenz
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Prof. Dr. Johannes Blömer

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Einführung in Kryptographie:* **Methodische Umsetzung** Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. **Lernmaterialien, Literaturangaben** Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography, Chapman and Hall, Johannes Buchmann: Einführung in Kryptographie, Springer Verlag, A Graduate Course in Applied Cryptography: https://crypto.stanford.edu/~dabo/cryptobook/, Vorlesungsfolien, Übungsaufgaben

Grundlagen der Algorithmische Geometrie (BA Inf v4)

Fundamentals of Computational Geometry (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2215	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05601 Grundlagen der Algorithmischen Geometrie	V3 Ü2	75	105	WP	40/20

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Grundlagen der Algorithmischen Geometrie:
Empfohlene Vorkenntnisse
Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse mit mathematischen mathematische Methoden zu erlernen. Grundkenntnisse einiger grundlegender Algorithmen und Datenstrukturen und deren Analyse wird angenommen.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Grundlagen der Algorithmischen Geometrie:
Es werden Algorithmen und Datenstrukturen aus dem Bereich der Algorithmischen Geometrie behandelt. Die Grundelemente der Eingabe sind geometrische Daten (Punkte, Linien, Kreise, Polygone, Körper). Die Probleme werden geometrisch formuliert und dafür wird eine algorithmische Lösung mit Hilfe spezieller geometrischer Datenstrukturen gesucht. Die Algorithmen werden theoretisch analysiert. Dazu wird Laufzeit und Speicherplatz bestimmt und die Korrektheit der Algorithmen bewiesen.

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden können grundlegende Techniken im Bereich der algorithmischen Geometrie anwenden. Sie können entscheiden, für welches geometrische Problem welcher Algorithmus am besten geeignet ist. Sie können die Algorithmen an eine neue Situation anpassen.
Nichtkognitive Kompetenzen:
• Haltung und Einstellung
• Lernkompetenz
• Schreib- und Lesekompetenz (wissenschaftlich)
• Selbststeuerungskompetenz</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Dr. Matthias Fischer</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Grundlagen der Algorithmischen Geometrie:* **Methodische Umsetzung** • Vorlesung mit Beamer und Tafelanschrieb • Übungen in Kleingruppen • erwartetet Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben • Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt • In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen vorgestellt. **Lernmaterialien, Literaturangaben** • Standardlehrbücher, Folien der Vorlesung, Übungsblätter • Computational Geometry: Algorithms and Applications, Mark de Berg, Otfried Cheong, Marc van Krefeld, Marc Overmars, Springer-Verlag, 2008 • Algorithmische Geometrie, Rolf Klein, Springer-Verlag, 2005 • Lectures on Discrete Geometry, Jiri Matousek, Springer Verlag, 2001

Grundlegende Algorithmen (BA Inf v4)

Fundamental Algorithms (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2211	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05511 Grundlegende Algorithmen	V3 Ü2	75	105	WP	100/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Grundlegende Algorithmen:
Empfohlene Vorkenntnisse
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.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Grundlegende Algorithmen:
In dieser Veranstaltung werden die folgenden algorithmischen Paradigmen untersucht, einschließlich genauer Analysen von Laufzeit und Korrektheit:

- Fortgeschrittene Suchstrukturen: Bereichssuche, Splay-Bäume, (a,b)-Bäume
- Prioritätswarteschlangen und Anwendungen: Fortgeschrittene Heap-Implementierungen, Anwendungen
- Graphenalgorithmen: Zusammenhangskomponenten, kürzeste Wege, Matchings
- Netzwerkflüsse: Algorithmus von Ford und Fulkerson, Preflow-Push-Algorithmus, Anwendungen
- Lineare Programmierung: Geometrische Interpretation, Dualität, Anwendungen
- String Matching-Algorithmen, Knuth-Morris-Pratt-Algorithmus, Boyer-Moore-Algorithmus

5 **Lernergebnisse und Kompetenzen:**

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. 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.
Nichtkognitive Kompetenzen:

- Haltung und Einstellung
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Prof. Dr. Sevag Gharibian

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Grundlegende Algorithmen:* **Methodische Umsetzung** • Vorlesung mit Beamer und Tafelanschrieb. • Übungen in Kleingruppen. • erwartete Aktivitäten der Studierenden: aktive Mitarbeit bei Präsenzübungen, Hausaufgaben. • Übungsblätter, Lösungen werden in Übungsgruppen vorgestellt und diskutiert. • In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt. **Lernmaterialien, Literaturangaben** Standardlehrbücher, Foliensatz der Vorlesung, Übungsblätter

Komplexitätstheorie (BA Inf v4)			
Complexity Theory (BA CS v4)			
Modulnummer: M.079.2212	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05512 Komplexitätstheorie	V3 Ü2	75	105	WP	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung 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.

- Komplexitätsklassen, P vs. NP
- Reduktionen und Vollständigkeit
- Platzkomplexität
- Hierarchiesätze
- Relativierung und Orakel-Turingmaschinen
- Polynomialzeit-Hierarchie
- Probabilistische Komplexitätsklassen

5 **Lernergebnisse und Kompetenzen:**

Studierende können die Eigenschaften wesentlicher Komlexitätsklassen 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.

Nichtkognitive Kompetenzen:

- Einsatz und Engagement
- Lernmotivation
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Prof. Dr. Johannes Blömer

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Komplexitätstheorie:* **Methodische Umsetzung** Eine Mischung aus Folien und Tafelanschrieb. Alle wichtigen Konzepte und Techniken werden in Übungen anhand von Beispielen weiter vertieft. **Lernmaterialien, Literaturangaben** • Michael Sipser, Introuction to the Theory of Computation, • S. Arora, B. Barak, Computational Complexity - A Modern Approach, Cambridge University Press, • Vorlesungsfolien, Übungsaufgaben

Quantum Networks (BA Inf v4)			
Quantum Networks (BA CS v4)			
Modulnummer: M.079.2230	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehr-form**	**Kontakt-zeit (h)**	**Selbst-studium (h)**	**Status (P/WP)**	**Gruppen-größe (TN)**
a)	L.079.05609 Quantum Networks	V3 Ü2	75	105	P	60/10

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Quantum Networks:
Empfohlene Vorkenntnisse
Kenntnisse aus dem Modul *Lineare Algebra für Informatik* sind hilfreich.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Quantum Networks:* Quantennetzwerke, in denen die Kraft der Quantenphysik genutzt werden kann um mehrere Quantenprozessoren über große Entfernungen zu verbinden, stehen an der vordersten Front der Quanteninformationswissenschaft. Sie ermöglichen weiterhin die Übertragung und Manipulation von Quanteninformationen in Form von Quantenbits oder Qubits, die Informationen auf eine Weise kodieren und verarbeiten können, die für klassische Bits unmöglich ist. Quantennetzwerke haben deshalb das Potenzial, neuartige Anwendungen wie verteiltes Quantenrechnen, Quantenkryptographie und sogar ein Quanteninternet zu ermöglichen. Quantennetzwerke stellen auch neue Herausforderungen und Möglichkeiten für Netzwerkdesign, Optimierung und Sicherheit dar. In Ihnen kommen Techniken aus Physik, Ingenieurwesen, Informatik und Mathematik zur Anwendung. Eines der Schlüsselkonzepte in Quantennetzwerken ist die Quantenverschränkung - ein Phänomen das auftritt wenn zwei oder mehr Qubits einen Quantenzustand teilen und nicht unabhängig voneinander beschrieben werden können, selbst wenn sie räumlich getrennt sind. Quantenverschränkung erlaubt die Erzeugung von Quantenkorrelationen und Quantenteleportation, die für die Quantenkommunikation und -berechnung wesentlich sind. Ein weiteres wichtiges Konzept in Quantennetzwerken sind Qudits, also Verallgemeinerungen von Qubits die mehr als zwei mögliche Zustände haben. Qudits können mehr Informationen pro physikalischem System als Qubits kodieren und ermöglichen neue Arten von Quantenoperationen und -protokollen. Diese Vorlesung führt in die Grundprinzipien und Anwendungen von Quantennetzwerken ein. Die Inhalte umfassen: • Verschränkung von Zwei- und Vielteilchensystemen • Quantenkommunikationsprotokolle und -techniken • Quantennetzwerkleistung im Vergleich zum klassischen Gegenstück • Quantennetzwerkarchitektur und -komponenten • Quantennetzwerkanwendungen und -herausforderungen
5	**Lernergebnisse und Kompetenzen:** Die Studierenden erlernen die modernsten Konzepte und Techniken im Bereich Quantennetzwerke, welcher Quanteninformationswissenschaft und Netzwerkwissenschaft verbindet. Diese Vorlesung vermittelt den Studierenden die Fähigkeiten und das Wissen, um Quantennetzwerke für verschiedene Anwendungen und Szenarien zu entwerfen, zu analysieren und zu implementieren. Nach Abschluss dieses Kurses werden die Studierenden in der Lage sein: • Quantennetzwerkarchitektur und -komponenten zu verstehen • Quantenkommunikationsprotokolle und -techniken zu implementieren • die Quantennetzwerkleistung und -optimierung zu bewerten • die Quantennetzwerksicherheit und -robustheit zu beurteilen

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	120-180 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Dr. Zahra Raissi

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Quantum Networks:
Methodische Umsetzung
Theoretische Grundlagen und Konzepte werden in Form von Vorlesungen vermittelt und in praktischen Übungen, Gruppenarbeiten sowie individuellen Hausaufgaben vertieft.
Lernmaterialien, Literaturangaben

- Michael A. Nielsen, Isaac L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.
- Ingemar Bengtsson and Karol Zyczkowski, Geometry of quantum states: an introduction to quantum entanglement, Cambridge university press, 2006, ISBN 9780511535048.
- Vorlesungsfolien
- Übungsaufgaben

Verteilte Algorithmen und Datenstrukturen (BA Inf v4)			
Distributed Algorithms and Data Structures (BA CS v4)			
Modulnummer: M.079.2214	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 6	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05610 Verteilte Algorithmen und Datenstrukturen	V3 Ü2	75	105	WP	60/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4	**Inhalte:** 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 selbststabilisierenden Systeme. Danach folgt eine kurze Einführung in verteilte Programmierung, damit die in der Vorlesung vorgestellten Datenstrukturen auch implementiert werden können. Anschließend werden zunächst prozessorientierte Datenstrukturen und dann informationsorientierte Datenstrukturen vorgestellt. • Einleitung • Netzwerktheorie • Designprinzipien für verteilte Algorithmen und Datenstrukturen • Einführung in verteilte Programmierung • Prozessorientierte Datenstrukturen: (Zyklische) Listen, de Bruijn Graphen, Skip Graphen, Delaunay Graphen • Informationsorientierte Datenstrukturen: Verteiltes Hashing, verteilte Queue, verteilter Stack, verteilter Heap *Inhalte der Lehrveranstaltung 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 implementiert werden können. Anschließend werden zunächst prozessorientierte Datenstrukturen und dann informationsorientierte Datenstrukturen vorgestellt. • Einleitung • Netzwerktheorie • Designprinzipien für verteilte Algorithmen und Datenstrukturen • Einführung in verteilte Programmierung • Prozessorientierte Datenstrukturen: (Zyklische) Listen, de Bruijn Graphen, Skip Graphen, Delaunay Graphen • Informationsorientierte Datenstrukturen: Verteiltes Hashing, verteilte Queue, verteilter Stack, verteilter Heap

5 **Lernergebnisse und Kompetenzen:**

- Kenntnis ausgewählter verteilter Algorithmen und Datenstrukturen
- Kenntnis wesentlicher Konzepte im Bereich verteilter Algorithmen und Datenstrukturen
- Fähigkeit, selbstständig adäquate Techniken und Verfahren im Bereich der verteilten Algorithmen und Datenstrukturen zu entwickeln
- Fähigkeit, algorithmische Probleme gemäß ihrer Lösbarkeit und Komplexität einzuschätzen
- Fähigkeit, grundlegende verteilte Datenstrukturen zu implementieren

Nichtkognitive Kompetenzen:

- Gruppenarbeit
- Lernkompetenz
- Schreib- und Lesekompetenz (wissenschaftlich)
- Selbststeuerungskompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12	**Modulbeauftragte/r:** Prof. Dr. Christian Scheideler
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Verteilte Algorithmen und Datenstrukturen:* **Methodische Umsetzung** Vorlesung mit Übungen, Hausaufgaben und Softwareprojekt **Lernmaterialien, Literaturangaben** Skript

Ausgewählte Themen im Gebiet Algorithmen und Komplexität			
Selected Topics in Algorithms and Complexity			
Modulnummer: M.079.2217	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5-6	**Dauer (in Sem.):** 1	**Sprache:** de / en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)

2 **Wahlmöglichkeiten innerhalb des Moduls:**

Innerhalb des Moduls ist genau eine der enthaltenen Lehrveranstaltungen zu wählen.

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

In diesem Modul werden wechselnde Lehrveranstaltungen, beispielsweise von Gastdozenten angeboten. Die konkreten Inhalte sind den Lehrveranstaltungen zu entnehmen.

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden sind in der Lage für ein spezifisches Themenfeld aus dem Gebiet Algorithmen und Komplexität:

- Problemstellungen adäquat zu beschreiben
- Problemstellungen zu analysieren und Lösungsansätze zu erarbeiten
- Lösungen zu gegebenen Problemstellungen zu erarbeiten und umzusetzen
- Lösungen zu gegebenen Problemstellungen zu bewerten
- Verschiedene Lösungen zu gegebenen Problemstellungen zu vergleichen, zu bewerten und eine adäquate Lösung auszuwählen

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
	Klausur, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

keine

8	**Voraussetzungen für die Teilnahme an Prüfungen:** Bestehen der Studienleistung
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Studiengangsbeauftragter Informatik
13	**Sonstige Hinweise:**

3.2 Computersysteme

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.

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.

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.

Einführung in Hochleistungsrechnen (BA Inf v4)

Introduction to High-Performance Computing (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2318	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05602 Einführung in Hochleistungsrechnen	V2 Ü3	75	105	WP	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3	**Teilnahmevoraussetzungen:** 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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt. *Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Hochleistungsrechnen:* Die Vorlesung baut auf den Inhalten der Lehrveranstaltungen “Rechnerarchitektur” und “Systemsoftware und systemnahe Programmierung” auf.
4	**Inhalte:** *Inhalte der Lehrveranstaltung Einführung in Hochleistungsrechnen:* Hochleistungsrechner sind eine zentrale Technologie für Modellierung, Simulation und Datenanalyse in zahlreichen wissenschaftlichen und technischen Anwendungen. Dieses Modul ist eine erste Einführung in das Hochleistungsrechnen und vermittelt einen Überblick über die Architektur, Leistungsbewertung, Programmierung und Optimierung paralleler Computersysteme. Im Fokus steht dabei die Erarbeitung eines übergreifenden Verständnisses von Hardwarearchitektur und Methoden der performance-orientierten Programmierung in Theorie und Praxis. Zur Vermittlung praktischer Fähigkeiten beinhaltet die Lehrveranstaltung auch einen erheblichen Anteil von Programmierübungen und Praktika auf Hochleistungsrechnern. Dabei werden sowohl Anwendungen aus dem Bereich der Computersimulation, wie auch aus dem Bereich der Datanalyse betrachtet. Als Programmiersprache kommt die Sprache Julia zum Einsatz, die moderne dynamische Programmierparadigmen mit hoher Performanz vereint und besonders für numerische Berechnungen, Datenanalyse und Hochleistungsrechnen geeignet ist. Die Veranstaltung umfasst folgende Inhalte: • Übersicht über die Architektur von Parallel- und Hochleistungsrechnern • Leistungsbewertung von parallelen Programmen und Hochleistungsrechnern • Modelle und Programmiermuster für parallele Programmierung • Optimierung serieller Programme • Vektorisierung • Parallelisierung mit gemeinsamem Speicher (Multi-threading) • Parallelisierung mit verteiltem Speicher • Anwendungen und Fallstudien • Beschleuniger (GPU)

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,
• die grundlegenden Ansätze zur Modellierung, Leistungsbewertung und Optimierung von Rechnersystemen zu erklären und auf Fallstudien anzuwenden,
• die Ebenen von Parallelverarbeitung in modernen Rechnersystemen zu benennen, ihre Eigenschaften zu charakterisieren und daraus Optimierungsstrategien abzuleiten,
• Programmiertechniken zur Parallelverarbeitung beim Entwurf und der Implementierung von Software praktisch anzuwenden,
• Parallele Anwendungen auf Hochleistungsrechnern auszuführen und eine Leistungsbewertung durchzuführen.
Nichtkognitive Kompetenzen:
• Gruppenarbeit
• Lernkompetenz
• Selbststeuerungskompetenz</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Christian Plessl</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Einführung in Hochleistungsrechnen:* **Methodische Umsetzung** In der Vorlesung werden die Grundlagen, Begriffe und Methoden des Hochleistungsrechnens vermittelt und mit Programmierbeispielen veranschaulicht. In den begleitenden Übungen werden die theoretischen Grundlagen vertieft und im begleitenden Praktikum in Programmierübungen anhand praktischer Fallstudien und kleinen Projekten in Kleingruppen selbst erprobt. **Lernmaterialien, Literaturangaben** • Vorlesungsfolien • Programmierbeispiele • Webseite

Einführung in Planungs- und Entscheidungsfindungsalgorithmen			
Introduction to Planning and Decision-Making Algorithms			
Modulnummer: M.048.11113	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5.-6. Semester	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.048.11113 Einführung in Planungs- und Entscheidungsfindungsalgorithmen	2V 2Ü, SS	60	120	P	30/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

Keine

3 **Teilnahmevoraussetzungen:**

Zwingend für WGBAET: Erfolgreicher Abschluss der nach Studienverlaufsplan im 1. und 2. Fachsemester abzuschließenden Module.
Andere Studiengänge: Keine

Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Planungs- und Entscheidungsfindungsalgorithmen:
Keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Einführung in Planungs- und Entscheidungsfindungsalgorithmen:
Die Vorlesung gibt eine Einführung in Algorithmen, die es Systemen ermöglicht, für uns Menschen in komplexen Umgebungen zu planen, in denen unsere eigene Wahrnehmung nicht genügt, um den Wert kleiner Schritte auf dem Weg zu einem weit entfernten Ziel vorauszusehen. Die Einführung in solche Algorithmen für diskrete Planungs- und Entscheidungsräume ist das Kernthema des vorliegenden Kurses. Der Kurs ist damit insbesondere für Studenten mit Interesse an Robotik, KI, Algorithmen und Computer Vision relevant.
Inhaltlich behandelt der Kurs die Grundlagen der diskreten Planungsalgorithmen, einschließlich:

- Diskrete suchbasierte Planung (Label Korrektur, Dijkstra, A*, etc.)
- Prinzip der dynamischen Programmierung
- Sequenzielle Entscheidungstheorie und entscheidungstheoretische Planung.
- Grundlagen des bestärkenden Lernens (Reinforcement Learning).
- Planung unter Unsicherheiten und partielle Beobachtbarkeit.
- Sampling-basierte Bewegungsplanung.

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:

Lernergebnisse und Kompetenzen:
Nach Abschluss des Moduls sind die Studierenden in der Lage:

• Suchbasierte Planungsalgorithmen zu analysieren und zu implementieren.
• Den Rechenaufwand von suchbasierten Planungsalgorithmen zu bewerten.
• Planungsprobleme als sequenzielle Entscheidungsprobleme zu formulieren.
• Sequenzielle Entscheidungsprobleme mit Reinforcement Learning zu lösen.
• Probleme zu verstehen, die sich aus der Planung unter unsicheren Informationen ergeben.
• Unsichere Planungsprobleme in Planungsprobleme in Informationsräumen umzuwandeln.

Fachübergreifende Kompetenzen:
Nach Abschluss des Moduls sind die Studierenden in der Lage:

• Erstellen von Simulationsumgebungen für Planungs- und Entscheidungsfindungsprobleme.
• Implementierung von Planungs- und Reinforcement Learning Algorithmen in Python.
• Anwendung von Wissen aus der System- und Wahrscheinlichkeitstheorie, um Algorithmen zu entwerfen und sequentielle Entscheidungsprobleme zu lösen.

Weiter haben die Studierenden ihre Kooperations- und Teamfähigkeit bei der Bearbeitung von Hausübungen erweitert.</td></tr>
<tr><td>6</td><td>Prüfungsleistung:

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>120-180 min oder 30-45 min</td><td>100%</td></tr>
</table>
</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:

keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:

Keine</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:

Die Vergabe der Leistungspunkte erfolgt, wenn die Modulabschlussprüfung (MAP) bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:

Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:

BF Automatisierungstechnik Lehramt BK affine Fächer Master v5, BF Elektrotechnik Lehramt BK Master v5, BF Informationstechnik Lehramt BK affine Fächer Master v5, Bachelorstudiengang Computer Engineering v3 (CEBA v3), Bachelorstudiengang Computer Engineering v3b (CEBA v3b), Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Elektrotechnik v6 (EBA v6), Bachelorstudiengang Elektrotechnik v7 (EBA v7), Bachelorstudiengang Informatik v4 (IBA v4), Bachelorstudiengang Informatik v5 (IBA v5), Bachelorstudiengang Wirtschaftsingenieurwesen Studienrichtung Elektrotechnik, Bachelorstudiengang Wirtschaftsingenieurwesen Studienrichtung Elektrotechnik V5</td></tr>
</table>

12	**Modulbeauftragte/r:** Dr. Adrian Redder
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Einführung in Planungs- und Entscheidungsfindungsalgorithmen:* **Lehrveranstaltungsseite:** `https://en.ei.uni-paderborn.de/de/rat` **Methodische Umsetzung:** • Tafelanschrieb im Wechsel mit kurzen Präsentationen und Simulationen. • Quizze zur Wiederholung. • Gruppenübungen (Theorie und Simulation). • Hausübungen (Theorie und Simulation) zum Erlangen von Bonuspunkten um die Modulnote um maximal 0,7 zu verbessern. **Literatur:** * Planning Algorithms, Steven M. LaValle, 2006. * Dynamic Programming and Optimal Control, Dimitri P. Bertsekas, vol. 1, 2012. * Vorlesungsskript

Eingebettete Systeme (BA Inf v4)

Embedded Systems (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2311	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05620 Eingebettete Systeme	V3 Ü2	75	105	WP	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Eingebettete Systeme:
Empfohlene Vorkenntnisse:
Kenntnisse aus der Lehrveranstaltung Rechnerarchitektur sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Eingebettete Systeme:
Die Veranstaltung bietet eine Einführung in Eingebettete Systeme und vermittelt Grundlagen zu Spezifikationsmodellen, eingebetteten Zielarchitekturen und Methoden zum Entwurf von reaktiven und Echtzeitsystemen sowie zur Bewertung und Analyse von Prozessor-Performance und -Energie.

- Einführung in Eingebettete Systeme
- Spezifikationsmodelle: Zustandsorientiert, Datenflussorientiert
- Zielarchitekturen: General-Purpose Prozessoren, Digitale Signalprozessoren, Mikrocontroller, ASIPs, FPGAs und ASICs, System-on-Chip
- Reaktive und Echtzeitsysteme: Tasksdefinitionen, Programmieransätze, Echtzeitscheduling, gemeinsam genutzte Ressourcen
- Performance und Energie: Worst-case execution time analysis, Energiemetriken, Techniken zur Energieminimierung

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden sind nach dem Besuch der Lehrveranstaltung in der Lage,

- die Eigenschaften eingebetteter Systeme zu benennen,
- Spezifikationsmodelle für eingebettete Systeme mit ihren Eigenschaften zu erklären,
- die Entwurfsziele und Eigenschaften wesentlicher Typen von eingebetteten Zielarchitekturen zu erklären,
- Ansätze zur Programmierung von Echtzeitsystemen aufzuzählen,
- Methoden zur Analyse von Echtzeiteigenschaften anzuwenden,
- Methoden zur Bestimmung der Worst-case execution time anzuwenden und
- die Bedeutung von Performance- und Energie-Metriken einzuschätzen.

Nichtkognitive Kompetenzen:

- Gruppenarbeit
- Lernkompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12	**Modulbeauftragte/r:** Prof. Dr. Marco Platzner
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Eingebettete Systeme:* **Methodische Umsetzung:** • Vorlesung mit Beamer und Tafelanschrieb • Interaktive Übungen im Hörsaal • Rechnerübungen mit eingebetteten Zielarchitekturen (DSP, ARM, FPGA) **Lernmaterialien, Literaturangaben:** • Vorlesungsfolien und Übungsblätter • Aufgabenblätter und technische Dokumentation für die Rechnerübungen • Peter Marwedel: Embedded System Design, Springer, 2011. • Aktuelle Hinweise auf alternative und ergänzende Literatur, sowie Lehrmaterialien auf der Webseite und in den Vorlesungsfolien

Embedded Machine Learning (BA Inf v4)

Embedded Machine Learning (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2331	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05648 Embedded Machine Learning	V2 Ü3	75	105	P	100/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Embedded Machine Learning:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen Programmierung 1, Rechnerarchitektur und Einführung in Data Science sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Embedded Machine Learning:
Eingebettetes maschinelles Lernen bezeichnet die Integration von Verfahren des maschinellen Lernens in eingebettete Systeme mit stark begrenzten Rechen-, Speicher- und Energieressourcen. Im Gegensatz zu cloudbasierten Ansätzen erfolgt die Datenverarbeitung und Inferenz lokal auf dem Zielsystem, beispielsweise auf Mikrocontrollern oder spezialisierten System-On-Chip Plattformen. Intelligente Funktionen werden direkt dort ausgeführt, wo Daten entstehen. Dadurch lassen sich Reaktionszeiten reduzieren, Energie sparen und Datenschutz erhöhen. Die wachsende Bedeutung des eingebetteten maschinellen Lernens ergibt sich aus der zunehmenden Verbreitung vernetzter Geräte und dem Bedarf an autonomer Entscheidungsfindung in Echtzeit. Die Veranstaltung vermittelt grundlegende Konzepte und Methoden des maschinellen Lernens für eingebettete Systeme und behandelt deren praxisnahe Umsetzung unter Einsatz aktueller Entwicklungswerkzeuge.

- Einführung in eingebettetes maschinelles Lernen
- Sensoren, Datenaufnahme und Signalvorverarbeitung
- Klassische Verfahren des maschinellen Lernens, z.B. k-Nearest Neighbor, Support Vector Machine, Decision Tree
- Neuronale Netzwerke und Deep Learning, z.B. Multilayer-Perceptron, Convolutional Neural Network
- Bibliotheken, Programmierschnittstellen und Frameworks für maschinelles Lernen, z.B. SiKit-Learn, PyTorch, TensorFlow, Keras
- Modellkompression: Pruning und Quantisierung
- Prozessoren und Hardwarebeschleuniger

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- den Aufbau und die Herausforderungen von Systemen des eingebetteten maschinellen Lernens benennen,
- die Funktionsweise von klassischen Verfahren des maschinellen Lernens und von tiefen neuronalen Netzen erklären,
- für gegebene Anwendungsfälle Verfahren des maschinellen Lernens auswählen und anwenden,
- Hardwareumgebungen und Software-Werkzeuge auswählen und anwenden,
- Methoden zur Modellkompression erklären und
- die Bedeutung von Parametern wie Genauigkeit der Modelle, Latenz und Durchsatz, Speicherbedarf und Energiebedarf einschätzen.

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Prof. Dr. Marco Platzner

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Embedded Machine Learning:
Methodische Umsetzung
Die Lehrveranstaltung gliedert sich in eine Vorlesung sowie in Rechen- und praktische Übungen. Die Vorlesung wird mittels Beamerpräsentationen und Tafelanschrieb durchgeführt. In den Rechenübungen werden Aufgaben gestellt, deren Lösungen im Rahmen der Übungseinheiten erarbeitet, vorgestellt und diskutiert werden. Die praktischen Übungen umfassen die Bearbeitung ausgewählter Aufgaben anhand konkreter Anwendungsbeispiele, die auf einem eingebetteten System mit einem 32-Bit-Mikrocontroller (STM32) und entsprechender Sensorik umgesetzt werden. Die Implementierung erfolgt in den Programmiersprachen Python und C.
Lernmaterialien, Literaturangaben

- Vorlesungsfolien, Aufgabenblätter für Rechenübungen
- Aufgabenblätter für Anwendungsbeispiele, technische Dokumentationen
- Cem Ünsalan, Berkan Höke, Eren Atmaca: Embedded Machine Learning with Microcontrollers, Springer, https://link.springer.com/book/10.1007/978-3-031-70912-8
- Ian Goodfellow and Yoshua Bengio and Aaron Courville: Deep Learning, MIT Press, https://www.deeplearningbook.org
- Andriy Burkov: The hundred-page machine learning book
- Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Verteilte Systeme (BA Inf v4)

Distributed Systems (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2313	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05533 Verteilte Systeme	V2 Ü3	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Verteilte Systeme:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Programmierung 1*, *Programmierung 2* und *Betriebssysteme* sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Verteilte Systeme:
Verteilte Systeme sind in unserem Alltag allgegenwärtig und unterstützen die von uns genutzten Dienste, ohne dass wir dies überhaupt bemerken. Von Cloud-Plattformen wie Google Drive und Apple iCloud über soziale Netzwerke wie Instagram und TikTok bis hin zu Kryptowährungen wie Bitcoin und sogar KI-Chatbots wie ChatGPT - sie alle basieren auf einer Reihe miteinander verbundener Maschinen, die als ein System zusammenarbeiten. In diesem Kurs werden Ihnen die Konzepte und Prinzipien des Aufbaus verteilter Systeme vorgestellt. Konkret behandeln wir die folgenden Themen:

- Kommunikation: Remote Procedure Call (RPC), Multicast
- Clocks: Clock-Synchronisierung, logische Uhren
- Global state and snapshots
- Replikation: Primary-Backup und replizierte State Machine
- Konsistenzmodelle: Eventualkonsistenz, kausale Konsistenz, sequenzielle Konsistenz, Linearizierbarkeit
- Fehlertoleranz: Paxos, Raft, Byzantinische Fehlertoleranz
- Transaktionen und Serializability
- Anwendungen: Distributed Hash Table, Blockchain

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden sind in der Lage,

- die grundlegenden Konzepte und Prinzipien verteilter Systeme zu erklären (Wissen und Verständnis)
- Protokoll-Details und -Eigenschaften anzuwenden, um Eigenschaften verteilter Systeme abzuleiten (Anwendung von Wissen und Verständnis)
- die Vorteile und Grenzen der heute gängigen verteilten Systeme zu erklären (Urteilsvermögen)
- ihre Kenntnisse in der Systemprogrammierung durch die Implementierung von Protokollen für verteilte Systeme unter Beweis zu stellen (Anwendung von Wissen und Verständnis)

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

<table>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Übungsaufgaben, schriftliche Ausarbeitung oder Referat</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Lin Wang</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Verteilte Systeme:
Methodische Umsetzung
Die Lehrveranstaltung besteht aus einer Vorlesung und begleitenden Übungen. Die Vorlesung erfolgt mit Beamer und Tafelanschrieb. In den Übungen werden sowohl konzeptionelle/analytische als auch praktische Aufgaben ausgegeben und von den Teilnehmern in Kleingruppen gelöst und anschließend diskutiert.
Lernmaterialien, Literaturangaben
• Vorlesungsfolien, Übungsblätter
• Standardlehrbücher: Distributed Systems (4th edition, Van Steen, Tanenbaum)
• Weitere Literatur wird in der Veranstaltung bekanntgegeben.</td></tr>
</table>

Ausgewählte Themen im Gebiet Computersysteme			
Selected Topics in Computer Systems (BA)			
Modulnummer: M.079.2317	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5-6	**Dauer (in Sem.):** 1	**Sprache:** de / en

1 Modulstruktur

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)

2 Wahlmöglichkeiten innerhalb des Moduls:

Innerhalb des Moduls ist genau eine der enthaltenen Lehrveranstaltungen zu wählen.

3 Teilnahmevoraussetzungen:

keine

4 Inhalte:

In diesem Modul werden wechselnde Lehrveranstaltungen, beispielsweise von Gastdozenten angeboten. Die konkreten Inhalte sind den Lehrveranstaltungen zu entnehmen.

5 Lernergebnisse und Kompetenzen:

Die Studierenden sind in der Lage für ein spezifisches Themenfeld aus dem Gebiet Computersysteme:

- Problemstellungen adäquat zu beschreiben
- Problemstellungen zu analysieren und Lösungsansätze zu erarbeiten
- Lösungen zu gegebenen Problemstellungen zu erarbeiten und umzusetzen
- Lösungen zu gegebenen Problemstellungen zu bewerten
- Verschiedene Lösungen zu gegebenen Problemstellungen zu vergleichen, zu bewerten und eine adäquate Lösung auszuwählen

6 Prüfungsleistung:

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
	Klausur, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 Studienleistung, qualifizierte Teilnahme:

keine

8	**Voraussetzungen für die Teilnahme an Prüfungen:** Bestehen der Studienleistung
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Studiengangsbeauftragter Informatik
13	**Sonstige Hinweise:**

3.3 Daten und Wissen

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.

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.

Angewandte Statistik mit R (BA Inf v4)

Applied Statistics with R (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2420	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05621 Angewandte Statistik mit R	V3 Ü2	75	105	WP	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3	**Teilnahmevoraussetzungen:** 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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt. *Teilnahmevoraussetzungen der Lehrveranstaltung Angewandte Statistik mit R:* **Empfohlene Vorkenntnisse** Grundlegende Kenntnisse und Interesse an Mathematik, Statistik und Wahrscheinlichkeitsrechnung sowie Basiswissen Programmierung
4	**Inhalte:** *Inhalte der Lehrveranstaltung Angewandte Statistik mit R:* Die Vorlesung gibt einen Überblick über sowohl deskriptive Statistik und explorative Datenanalyse als auch induktive Statistik und Grundlagen der Wahrscheinlichkeitsrechnung. Der Schwerpunkt liegt neben der Vermittlung von intuitivem Verständnis auf der angewandten Datenanalyse unter Zuhilfenahme der statistischen Programmiersprache R, die im ersten Block der Vorlesung im Rahmen eines Programmierkurses eingeführt wird, entsprechende Vorkenntnisse sind hier nicht erforderlich. Die Veranstaltung umfasst folgende Inhalte: • Einführung in die Programmiersprache R • Deskriptive Statistik und explorative Datenanalyse • Wahrscheinlichkeitsrechnung und statistische Verteilungen • Induktive Statistik, Konfidenzintervalle, statistisches Testen • Einführung in überwachtes (maschinelles) Lernen: Regressionsanalyse • Einführung in unüberwachtes (maschinelles) Lernen: Clustering
5	**Lernergebnisse und Kompetenzen:** Nach Abschluss des Moduls können die Studierenden . . . • die Grundprinzipien deskriptiver und induktiver Statistik sowie der Wahrscheinlichkeitsrechnung verstehen • statistische Techniken kompetent anwenden • die Statistik-Software R zur statistischen Datenanalyse und Visualisierung anwenden • Problemstellungen im Team analysieren und praxisrelevante Lösungen präsentieren
6	**Prüfungsleistung:** ☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

<table>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Heike Trautmann</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Angewandte Statistik mit R:
Methodische Umsetzung
Eine Einführung in die statistische Programmiersprache R wird kompakt in den ersten Wochen der Veranstaltung durchgeführt. Statistische Methoden werden durch inhaltliche Präsentation und interaktive Übung im Rahmen der Vorlesung vermittelt sowie in Tutorien vertieft, sowohl methodisch formal als auch anwenungsorientiert unter Verwendung von R.
Lernmaterialien, Literaturangaben
Für die statistische Programmiersprache R wird empfohlen:
Hadley Wickham & Garrett Grolemund (2023). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. 2nd ed. O’Reilly
Torsten Hothorn and Brian S. Everitt (2014). A Handbook of Statistical Analyses Using R. Chapman & Hall/CRC Press, 3rd edition, 2014.
C. Heumann, M. Schomaker, and Shalabh. Introduction to Statistics and Data Analysis With Exercises, Solutions and Applications in R. Springer, 2017.
Die methodischen Kapitel basieren auf der Kombination einer Vielzahl von Literaturquellen, diese werden in der Vorlesung bekannt gegeben.</td></tr>
</table>

Databases and Information Systems (BA Inf v4)			
Databases and Information Systems (BA CS v4)			
Modulnummer: M.079.2412	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05532 Databases and Information Systems	V3 Ü2	75	105	WP	120/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Databases and Information Systems:
Empfohlene Vorkenntnisse
Studierende sollten Vorkenntnisse in relationalen Datenbanken und SQL haben, die in etwa denen der Vorlesung “Datenbanksysteme” entsprechen, sowie Vorkenntnisse im Programmieren, die in etwas denen der Vorlesungen “Programmierung” und “Grundlagen der Programmiersprachen” entsprechen.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Databases and Information Systems:
Datenspeicherung und Datenmanagement spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissens von Unternehmen in Daten abgelegt ist. Zudem wachsen die Mengen gespeicherter Daten ständig, und eine Verarbeitung dieser riesigen Datenmengen erfordert Kenntnisse, die über SQL und traditionelle Datenbanken hinausgehen. Beispiele für diese riesigen Datenmengen sind Genomdatenbanken, Textdokumentsammlungen, Sensordaten, Satellitendaten, Daten aus Kameras, Mikrofonen, oder RFID-Tags, Telekommunikationsdaten, Wetterdaten, Finanzdaten, Newsreader, Daten aus Messenger-Diensten, etc.. Anwendungen oder Informationssysteme zu entwickeln, die bei diesen Datenmengen akzeptable Antwortzeiten haben, erfordert Kenntnisse über Nicht-Standard-Datenmodelle, Hauptspeicher-Datenbanken, Kompression, Indizierung riesiger Datenbestände und effiziente Suche auf diesen Datenbeständen.
Dieses Modul behandelt schwerpunktmäßig Algorithmen zur Kompression und zur effizienten Verarbeitung von komplexen, strukturierten Massendaten einschließlich Textdaten, Genomdaten, baumstrukturierter Daten und Graph-Daten. Zu den Inhalten gehört:

- Überblick über Suchmaschinen und Informationssysteme
- Hauptspeicherdatenbanken und Succinct-Codierungstechniken
- String-Kompressionsalgorithmen
- Genom-Datenbanken
- Verarbeitung riesiger Baum-Datenbestände (XML und JSON) und Baum-Kompression
- Graph-Datenbanken und Graph-Kompression
- Suchalgorithmen für Big Data und für Datenströme

5 **Lernergebnisse und Kompetenzen:**

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 unkomprimierten 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.
Nichtkognitive Kompetenzen:

- Gruppenarbeit
- Lernkompetenz
- Lernmotivation

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

<table>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Stefan Böttcher</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Databases and Information Systems:
Methodische Umsetzung
Grundlegende Konzepte werden in einer Vorlesung präsentiert. Zusätzlich werden theoretische Konzepte in Tutorien in Kleingruppen vertieft, insbesondere für Kernkonzepte von Datenbanken wie die Suche in und Anfragen auf Big Data, verteilten Datenbanken und mobilem Datenmanagement. Zudem erwerben Studierende praktische Kenntnisse durch Computer-gestützte Übungen, in denen sie aufbauend auf den in der Vorlesung erläuterten Konzepten ihre eigenen Informatonssysteme, Such- oder Kompressionsalgrithmen entwickeln.
Lernmaterialien, Literaturangaben
Verweise auf aktuelles Lernmaterial werden in der Vorlesung gegeben.</td></tr>
</table>

Embedded Machine Learning (BA Inf v4)

Embedded Machine Learning (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2331	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05648 Embedded Machine Learning	V2 Ü3	75	105	P	100/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Embedded Machine Learning:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen Programmierung 1, Rechnerarchitektur und Einführung in Data Science sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Embedded Machine Learning:
Eingebettetes maschinelles Lernen bezeichnet die Integration von Verfahren des maschinellen Lernens in eingebettete Systeme mit stark begrenzten Rechen-, Speicher- und Energieressourcen. Im Gegensatz zu cloudbasierten Ansätzen erfolgt die Datenverarbeitung und Inferenz lokal auf dem Zielsystem, beispielsweise auf Mikrocontrollern oder spezialisierten System-On-Chip Plattformen. Intelligente Funktionen werden direkt dort ausgeführt, wo Daten entstehen. Dadurch lassen sich Reaktionszeiten reduzieren, Energie sparen und Datenschutz erhöhen. Die wachsende Bedeutung des eingebetteten maschinellen Lernens ergibt sich aus der zunehmenden Verbreitung vernetzter Geräte und dem Bedarf an autonomer Entscheidungsfindung in Echtzeit. Die Veranstaltung vermittelt grundlegende Konzepte und Methoden des maschinellen Lernens für eingebettete Systeme und behandelt deren praxisnahe Umsetzung unter Einsatz aktueller Entwicklungswerkzeuge.

- Einführung in eingebettetes maschinelles Lernen
- Sensoren, Datenaufnahme und Signalvorverarbeitung
- Klassische Verfahren des maschinellen Lernens, z.B. k-Nearest Neighbor, Support Vector Machine, Decision Tree
- Neuronale Netzwerke und Deep Learning, z.B. Multilayer-Perceptron, Convolutional Neural Network
- Bibliotheken, Programmierschnittstellen und Frameworks für maschinelles Lernen, z.B. SiKit-Learn, PyTorch, TensorFlow, Keras
- Modellkompression: Pruning und Quantisierung
- Prozessoren und Hardwarebeschleuniger

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- den Aufbau und die Herausforderungen von Systemen des eingebetteten maschinellen Lernens benennen,
- die Funktionsweise von klassischen Verfahren des maschinellen Lernens und von tiefen neuronalen Netzen erklären,
- für gegebene Anwendungsfälle Verfahren des maschinellen Lernens auswählen und anwenden,
- Hardwareumgebungen und Software-Werkzeuge auswählen und anwenden,
- Methoden zur Modellkompression erklären und
- die Bedeutung von Parametern wie Genauigkeit der Modelle, Latenz und Durchsatz, Speicherbedarf und Energiebedarf einschätzen.

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

<table>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Übungsaufgaben, schriftliche Ausarbeitung oder Referat</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Marco Platzner</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Embedded Machine Learning:
Methodische Umsetzung
Die Lehrveranstaltung gliedert sich in eine Vorlesung sowie in Rechen- und praktische Übungen. Die Vorlesung wird mittels Beamerpräsentationen und Tafelanschrieb durchgeführt. In den Rechenübungen werden Aufgaben gestellt, deren Lösungen im Rahmen der Übungseinheiten erarbeitet, vorgestellt und diskutiert werden. Die praktischen Übungen umfassen die Bearbeitung ausgewählter Aufgaben anhand konkreter Anwendungsbeispiele, die auf einem eingebetteten System mit einem 32-Bit-Mikrocontroller (STM32) und entsprechender Sensorik umgesetzt werden. Die Implementierung erfolgt in den Programmiersprachen Python und C.
Lernmaterialien, Literaturangaben
• Vorlesungsfolien, Aufgabenblätter für Rechenübungen
• Aufgabenblätter für Anwendungsbeispiele, technische Dokumentationen
• Cem Ünsalan, Berkan Höke, Eren Atmaca: Embedded Machine Learning with Microcontrollers, Springer, https://link.springer.com/book/10.1007/978-3-031-70912-8
• Ian Goodfellow and Yoshua Bengio and Aaron Courville: Deep Learning, MIT Press, https://www.deeplearningbook.org
• Andriy Burkov: The hundred-page machine learning book
• Weitere Literatur wird in der Veranstaltung bekanntgegeben.</td></tr>
</table>

Foundations of the Semantic Web (BA Inf v4)

Foundations of the Semantic Web (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2414	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05630 Foundations of the Semantic Web	V2 Ü3	75	105	WP	24

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Foundations of the Semantic Web:
Empfohlene Vorkenntnisse
Logik, Java

4 **Inhalte:**

Inhalte der Lehrveranstaltung Foundations of the Semantic Web:
Die Lehrveranstaltung setzt sich mit der Repräsentation von maschinenlesbarem Wissen im Web auseinander. Nach der Vermittlung von Grundlagen zum Web werden die Syntax und formale Semantik der Sprachen RDF, RDFS, OWL, SPARQL untersucht. Außerdem werden Algorithmen zur Wissensextraktion und -integration vorgestellt. Praktische Anwendungsbeispiele für Technologien des Semantic Web runden den Inhalt der Veranstaltung ab.

- RDF Syntax und Semantik
- RDFS Syntax und Semantik
- OWL Syntax und Semantik
- SPARQL Syntax und Semantik
- Wissensextraktion
- Verknüpfung von Ressourcen
- Linked Data
- Anwendungen

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden verstehen die Relevanz des Semantic Web und können relevante Techniken zur Darstellung von Wissen beschreiben und auswählen. Sie können vorhandene Techniken beschreiben, um einfache Wissensgraphen aus unstrukturierten und halbstrukturierten Datenquellen zu extrahieren. Sie verstehen die Konzepte der Wissensintegration, können Alternativen benennen und ein für ein bestimmtes Problem geeignetes Verfahren auswählen. Auf dieser Grundlage können die Studierenden wissensbasierte Anwendungen und deren Lebenszyklen entwerfen, realisieren und evaluieren.

Nichtkognitive Kompetenzen:

- Lernkompetenz
- Medienkompetenz
- Schreib- und Lesekompetenz (wissenschaftlich)
- Gruppenarbeit

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Schriftliche Übungsaufgaben		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12 **Modulbeauftragte/r:**

Prof. Dr. Axel-Cyrille Ngonga Ngomo

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Foundations of the Semantic Web:* **Methodische Umsetzung:** In den wöchentlichen Vorlesungen (2 SWS) werden neue Inhalte behandelt. Neben formalen Betrachtungen werden auch Anwendungen und entsprechende Einschränkungen der im Laufe des Kurses vorgestellten Sprachen und Methoden behandelt. Die Übungen (1 SWS) sind sowohl theoretischer als auch praktischer Natur. Die Lernenden sollen zeigen, dass sie die Konzepte verstanden haben und sie auf praktische Probleme anwenden können. Das Miniprojekt (2 SWS) vermittelt den Studierenden eine ganzheitliche Sicht, wie ein komplexes Problem mit Hilfe von Technologien des Semantic Web gelöst werden kann. **Lernmaterialien, Literaturangaben:** Folien, Übungsaufgaben

Introduction to Knowledge Representation and Reasoning (BA Inf v4)

Introduction to Knowledge Representation and Reasoning (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2431	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05645 Introduction to Knowledge Representation and Reasoning	V3 Ü2	75	105	P	35

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Introduction to Knowledge Representation and Reasoning:
Empfohlene Vorkenntnisse

- gute Kenntnisse zu Aussagenlogik und grundlegendes Wissen zu Prädikatenlogik
- Kenntniss der Methoden zum Erfüllbarkeitstest in Aussagenlogik (aus dem Modul “Modellierung”)
- Kenntnis zu Entscheidbarbarkeit und der grundlegenden Komplexitätsklassen (z.B. P, NP, PSpace)

4 **Inhalte:**

Inhalte der Lehrveranstaltung Introduction to Knowledge Representation and Reasoning:
Dieses Modul führt grundlegende Formalismen der Logik-basierten Wissenrepräsentation ein:

- Defaultlogik
- Datalog
- Beschreibungslogiken.

Es werden zudem verschiedentliche Schlussfolgerungsprobleme eingeführt und deren Eigenschaften analysiert:

- nicht-monotones Schliessen,
- Regel-basiertes Schliessen
- Methoden zum Lernen von Logikformeln aus Beispielen.

Die Vorlesung behandelt auch Formalismen, die als Standards in der Praxis relevant sind, wie

- OWL
- SHACL

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- erklären was einen Logik-basierten Formalismus charakterisiert
- Grundbegriffe der Wissenrepräsentation kompetent verwenden und erklären
- elementare Formalismen der Wissensrepräsentation anwenden
- elementare Inferenzprobleme lösen
- Methoden zur Analyse von Schlussfolgerungsproblemen anwenden und erklären

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8	**Voraussetzungen für die Teilnahme an Prüfungen:** Bestehen der Studienleistung
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4)
12	**Modulbeauftragte/r:** Prof. Dr.-Ing. Anni-Yasmin Turhan
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Introduction to Knowledge Representation and Reasoning:* Dieses Modul besteht aus Vorlesung und Übung. Die Vorlesung führt die Begriffe, Methoden und Resultate ein. Die Übungsgruppen dienen der praktischen Anwendung von Gelerntem sowie Klärung von Fragen. Prüfungsrelevante Lernmaterialien sind der Tafelanschrieb und die Lösungen der Übungsblätter. Weiterführende Literatur wird in der Veranstaltung und im Panda-Kurs bekannt gegeben.

Knowledge Base Engineering (BA Inf v4)			
Knowledge Base Engineering (BA CS v4)			
Modulnummer: M.079.2430	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05514 Knowledge Base Engineering	V3 Ü2	75	105	P	35

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

Teilnahmevoraussetzungen der Lehrveranstaltung Knowledge Base Engineering:
Empfohlene Vorkenntnisse
Kenntnisse zu Prädikatenlogik und den grundlegenden Komplexitätsklassen ist sehr hilfreich, sowie Kenntnisse aus dem Modul “Modellierung”. Kenntnisse der Grundlagen aus den Modulen “Berechenbarkeit und Komplexität” und “Foundations of the Semantic Web” sind wünschenswert.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Knowledge Base Engineering:
Diese Vorlesung behandelt Verfahren zum Erstellen und Warten von logik-basierten Wissensbasen. Diese Art von Wissensbasen und ihre Schlußfolgerungsverfahren sind Grundlage für symbolisch Künstliche Intelligenz und auch für Semantische Technologien. Es werden in dieser Vorlesung Wissensbasen in der Form von Ontologien mit formaler Semantik betrachtet. Die Veranstaltung umfasst folgende Inhalte:

- Verfahren zur Wissensgewinnung für den Aufbau von Ontologien
- Einführung in wichtige Beschreibungslogiken und deren Schlussfolgerungsverfahren
- Ontologie-Standards: OWL 2 und RDFS
- Verfahren zum Lernen von Ontologien
- Verfahren für Erklärungen von Ergebnissen aus Schlussfolgerungsverfahren
- Verfahren zur Ontologie-Integration und zur Ontologie-Modularisierung

5 Lernergebnisse und Kompetenzen:

Die Studierenden können

- erklären was eine (OWL) Ontologie ist
- eine Ontologie erzeugen
- Ontologien verbinden
- Ontologien testen und berichtigen
- Logik-basierte Techniken für diese Verfahren erklären und anwenden

6 Prüfungsleistung:

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8 Voraussetzungen für die Teilnahme an Prüfungen:

Bestehen der Studienleistung

9 Voraussetzungen für die Vergabe von Credits:

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 Gewichtung für Gesamtnote:

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 Verwendung des Moduls in anderen Studiengängen:

Bachelorstudiengang Informatik v4 (IBA v4)

12 Modulbeauftragte/r:

Prof. Dr.-Ing. Anni-Yasmin Turhan

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Knowledge Base Engineering:* **Methodische Umsetzung** Dieses Modul besteht aus Vorlesung und Übung. Die Vorlesung führt die Begriffe, Methoden und Resultate ein. Die Übungsgruppen dienen der praktischen Anwendung von Gelerntem sowie Klärung von Fragen. Die Beteiligung bei der Übung sind Grundlage für die Studienleistung. Prüfungsrelevante Lernmaterialien sind die im Panda-Kurs bereitgestellten Foliensätze und Übungsblätter. Weiterführende Literatur wird in der Veranstaltung und im Panda-Kurs bekannt gegeben.

Ausgewählte Themen im Gebiet Daten und Wissen

Selected Topics in Data and Knowledge (BA)

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2417	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5-6	1	de / en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)

2 **Wahlmöglichkeiten innerhalb des Moduls:**

Innerhalb des Moduls ist genau eine der enthaltenen Lehrveranstaltungen zu wählen.

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

In diesem Modul werden wechselnde Lehrveranstaltungen, beispielsweise von Gastdozenten angeboten. Die konkreten Inhalte sind den Lehrveranstaltungen zu entnehmen.

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden sind in der Lage für ein spezifisches Themenfeld aus dem Gebiet Daten und Wissen:

- Problemstellungen adäquat zu beschreiben
- Problemstellungen zu analysieren und Lösungsansätze zu erarbeiten
- Lösungen zu gegebenen Problemstellungen zu erarbeiten und umzusetzen
- Lösungen zu gegebenen Problemstellungen zu bewerten
- Verschiedene Lösungen zu gegebenen Problemstellungen zu vergleichen, zu bewerten und eine adäquate Lösung auszuwählen

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	Prüfungsform	Dauer bzw. Umfang	Gewichtung für die Modulnote
	Klausur, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

keine

8	**Voraussetzungen für die Teilnahme an Prüfungen:** Bestehen der Studienleistung
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Studiengangsbeauftragter Informatik
13	**Sonstige Hinweise:**

3.4 Softwaretechnik

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 Handwerkszeug beherrschen, um sich im späteren Berufsleben in zukünftige Techniken einzuarbeiten.

Applied User Experience Design (BA Inf v4)

Applied User Experience Design (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2131	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	6	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05514 Applied User Experience Design	V2 Ü3	75	105	P	60/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Applied User Experience Design:* Die Studierenden wenden den theoretischen Hintergrund aus der Vorlesung “Gestaltung von Nutzungsschnittstellen/User Interface Design” an, um Interfaces zu entwerfen, die benutzbar und praxisnah sind, und um praktische Erfahrungen im Design und der Entwicklung von Prototypen eines Interfaces zu sammeln. Während des Semesters durchlaufen die Studierenden einen kompletten Designzyklus – Produktentstehung, explorative User Research, Ideenfindung, Prototyping und Bewertung – und stützen sich dabei auf Prinzipien aus dem UX-Design und Design Thinking. Die Studierenden lernen, Lösungen zu entwickeln, die einen Ausgleich zwischen pragmatischem Wert (Usability und Utility) und hedonistischen Eigenschaften (optische Attraktivität und Freude) schaffen. Hauptfokus des User Experience Designs ist das Verständnis für Nutzende und Stakeholder. Da Forschung für diesen Prozess von entscheidender Bedeutung ist, konzentriert sich dieser Kurs auf den Aufbau von Forschungskompetenzen durch User Research und Nutzerfeedback. Darüber hinaus vermittelt dieser Kurs weiterführende konzeptionelle Grundlagen, die den Weg für weitere Kurse und spätere Forschungen im Bereich der Mensch-Maschine-Interaktion ebnen, einschließlich der Schnittstelle zwischen empirischer Softwareentwicklung und IT-Sicherheit. Der Kurs umfasst folgende Inhalte: • Einführung in das UX-Design • Einführung in die UX-Forschung/Human Factors Research/Mental Models • Einführung in Design-Frameworks (Design Thinking, Partizipatives Design, Nutzerzentriertes Design) • Empathie-Mapping-Framework + Service Blue Printing • Design in der Softwareentwicklung, wie interagieren Designer und Entwickler • Prototyping und Feedback, Evaluierungsmethoden
5	**Lernergebnisse und Kompetenzen:** Die Studierenden sind in der Lage • die Anforderungen ihrer Stakeholder durch User Research empirisch und einfühlsam zu ermitteln. • diese Anforderungen in Designlösungen umzusetzen, indem sie auf kreative und konstruktive Weise Konzepte entwickeln. • die theoretischen Grundlagen aus der Vorlesung „Gestaltung von Nutzungsschnittstellen/User Interface Design” anzuwenden, um Schnittstellen zu schaffen, die sowohl hedonisch als auch pragmatisch sind. • praktische Erfahrungen im Entwerfen, Entwickeln und Verfeinern von Prototypen von Schnittstellen zu sammeln, einschließlich deren Bewertung durch Nutzenden-Feedback und Tests.

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min oder 30-45 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Fortschrittsberichte</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Yasemin Acar</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Applied User Experience Design:* **Methodische Umsetzung** Der Kurs beinhaltet eine Design-Challenge und die Studierenden müssen ein Designprojekt durchführen, bei dem sie den kompletten nutzerzentrierten Designzyklus kennenlernen. Die Studierenden werden in Gruppen (3-4) an einem Designproblem arbeiten. Sie müssen einen funktionalen Prototyp mit Schwerpunkt auf User Experience präsentieren. Im Rahmen des Projekts lernen die Studierenden, Informationsarchitektur, Accessibility, integratives Design und ethische Aspekte aus dem EthicalOS-Framework zu berücksichtigen. Wir werden uns wöchentlich treffen; es wird Präsentationen geben, die Studierendengruppen werden ebenfalls wöchentlich ihre Arbeit und ihren Fortschritt präsentieren. **Lernmaterialien, Literaturangaben** • Vorlesungsaufzeichnungen, Folien (auf Englisch) • Hassenzahl, Marc (2010): Experience Design: Technology for All the Right Reasons. Morgan and Claypool Publishers. • Norman, D. A. (1988). The Design of everyday things. • Costanza-Chock, Sasha(2020). Design justice: Community-led practices to build the worlds we need. The MIT Press. • Monteiro, Mike (2019). Ruined by design: How designers destroyed the world, and what we can do to fix it. Mule Design. • Mitrović, I., Auger, J., Hanna, J., & Helgason, I. (2021). Beyond speculative design: past–present–future. University of Split. • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Modellbasierte Softwareentwicklung (BA Inf v4)			
Model-Based Software Engineering (BA CS v4)			
Modulnummer: M.079.2114	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05543 Modellbasierte Softwareentwicklung	V3 Ü2	75	105	WP	75/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung 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.

- Software-Modelle, u.A. mit UML
- Views und Viewpoints
- Metamodellierung
- Statische und Dynamische Semantik
- Modelltransformationen
- Softwarearchitekturen
- Praxiseinsatz der modellbasierten und modellgetriebenen Softwareentwicklung
- aktuelle Trends und Forschungsthemen in der modellbasierten und modellgetriebenen Softwareentwicklung

5 **Lernergebnisse und Kompetenzen:**

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.
Nichtkognitive Kompetenzen:

- Einsatz und Engagement
- Lernkompetenz

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 40 min	100%

Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.

<table>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table>
<tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr>
<tr><td>a)</td><td>Schriftliche Übungsaufgaben</td><td></td><td>SL</td></tr>
</table>
Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Dr. Enes Yigitbas</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Modellbasierte Softwareentwicklung:
Methodische Umsetzung
Vorlesung mit Beamer und praktische Rechnerübungen.
Lernmaterialien, Literaturangaben

Ghezzi: Fundamentals of Software Engineering (Addison Wesley)
Reiko Heckel, Gabriele Taentzer: Graph Transformation for Software Engineers - With Applications to Model-Based Development and Domain-Specific Language Engineering. Springer 2020
Marco Brambilla, Jordi Cabot, Manuel Wimmer: Model-Driven Software Engineering in Practice, Second Edition. Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers 2017
</td></tr>
</table>

Secure Software Engineering (BA CS v4)

Secure Software Engineering (BA CS v4)

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2115	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5	1	en

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05640 Secure Software Engineering	V3 Ü2	75	105	WP	100

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Secure Software Engineering:
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.

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

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
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.
Nichtkognitive Kompetenzen:
• Lernkompetenz
• Lernmotivation</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Eric Bodden</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Secure Software Engineering:
Lernmaterialien, Literaturangaben:
Gary McGraw: Software Security: Building Security In (2006, Addison-Wesley Professional). Über UPB als e-book verfügbar.</td></tr>
</table>

Systematische Entwicklung von VR/AR Anwendungen (BA Inf v4)			
Systematic Development of VR/AR Applications (BA CS v4)			
Modulnummer: M.079.2118	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: null	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05604 Systematische Entwicklung von VR/AR Anwendungen	V3 Ü2	75	105	WP	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Systematische Entwicklung von VR/AR Anwendungen:* Obwohl die Begriffe Virtual Reality (VR) und Augmented Reality (AR) oftmals noch immer mit Unterhaltung und Videospielen assoziiert werden, haben sich die vielfältig nutzbaren Anwendungsbereiche der Technologien auch in anderen Branchen wie der Industrie, der Bildung oder auch der Medizin herausgestellt, wodurch deren Einsatz seit Langem Thema aktueller Forschung ist. Während Virtual Reality die Vorstellung einer virtuellen Welt realisiert, in der sich der Anwender durch die Manipulation seiner Sinne präsent fühlt und vollkommen in eine virtuelle Welt eintaucht, ohne jeglichen Bezug zur Außenwelt zu haben, ist Augmented Reality eine Erweiterung der Realität. Das Erwerben von und interagieren mit digitalen Daten in einem realen Raum steht hier im Vordergrund. Die Vorlesung führt grundlegende Technologien und Paradigmen der Virtual Reality (VR) und Augmented Reality (AR) ein. In den ersten Wochen lernen die Studierenden Methoden des Trackings, der 3D/2D Anzeige und Interaktion mit virtuellen Objekten sowie aktuelle Anzeigegeräte (z.B. Microsoft Hololens oder Meta Quest) und Software (z.B. Unity) kennen. Die erlernten Konzepte, Methoden und Technologien werden in praktischen Übungsaufgaben vertieft. Zusätzlich werden die Studierenden in Kleingruppen ein Mini-Forschungsprojekt im Themenkomplex VR/AR planen und umsetzen. Die Studierenden erarbeiten dabei auf Basis der aktuellen Forschungslage eine Forschungsfrage, die sie mit Hilfe des zu entwickelnden Prototypen beantworten möchten. Inhalt: • Einführung, Entwicklung, Begriffsdefinitionen • VR/AR-Hardware (Ein- und Ausgabegeräte) • Wahrnehmungsaspekte von VR/AR • Computergrafik von virtuellen Welten • Tracking • Anzeige, Interaktion und Navigation • Immersion • Modellierung und Annotation • Software-Architekturen • Echtzeit-Aspekte • Evaluation von VR/AR Anwendungen • VR/AR Systeme aus der Forschung
5	**Lernergebnisse und Kompetenzen:** Das Ziel dieser Vorlesung ist die Vermittlung von grundlegenden Methoden zur systematischen Erstellung und Darstellung virtueller Umgebungen. Die Studierenden sind in der Lage Virtual Reality (VR) und Augmented Reality (AR) - Systeme zu analysieren und zu evaluieren und kennen die verschiedenen Technologien und deren Probleme. Sie haben fortgeschrittene Programmierkenntnisse zur Erstellung von VR und AR Szenen u.a. mittels Unity3D. Es werden unter anderem Virtual-Reality- und Augmented-Reality-Anwendungen aus wissenschaftlich-technischen und industriellen Gebieten vorgestellt. **Nichtkognitive Kompetenzen:** • Einsatz und Engagement • Lernkompetenz • Medienkompetenz • Gruppenarbeit

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Schriftliche Übungsaufgaben oder Mitarbeit in Kleinprojekt</td><td></td><td>SL</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Bestehen der Studienleistung</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Dr. Enes Yigitbas</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Systematische Entwicklung von VR/AR Anwendungen:* **Methodische Umsetzung** • Vorlesung mit praktischen Übungsaufgaben • Mini-Forschungsprojekt in Kleingruppen zur Erstellung und Evaluation von VR/AR Anwendungen **Lernmaterialien, Literaturangaben** • Ralf Dörner, Wolfgang Broll, Paul Grimm, Bernhard Jung (Hrsg.): Virtual und Augmented Reality (VR/AR). Springer Vieweg, 2013. • William R. Sherman, Alan B. Craig: Understanding Virtual Reality – Interface, Application, and Design. Elsevier Science, 2003. • Dieter Schmalstieg, Tobias Höllerer: Augmented Reality – Principles and Practice. Addison-Wesley, 2016. • Joseph J. LaVioal, Ernst Kruijff, Ryan P. Mc Mahan, Doug A. Bowman, Ivan Poupyrev: 3D User Interfaces – Theory and Practice. Pearson Education, 2017. • Toni Parisi. Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile. O'Reilly, 2015. • Jonathan Linowes: Unity Virtual Reality Projects – Explore the world of virtual reality by building immersive and fun VR projects using Unity 3D. Packt Publishing, 2015.

Systems Engineering (BA Inf v4)			
Systems Engineering (BA CS v4)			
Modulnummer: M.079.2119	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05504 Systems Engineering	V2 Ü3	75	105	P	20

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Systems Engineering:
Maschinelle Intelligenz, Industrie 4.0 – dahinter verbergen sich Intelligente Technische Systeme (ITS): Komplexe Produkte aus dem Zusammenspiel von Ingenieurwissenschaften und Informations- und Kommunikationstechnologien. Die Entwicklung von ITS stellt Maschinenbauer, Informatiker und Wirtschaftsingenieure vor neue Herausforderungen, wird aber in Zukunft immer bedeutender sein.
Die Vorlesung vermittelt zunächst ein umfassendes Verständnis von ITS und erläutert den Wandel von der Mechatronik hin zu Systemen mit inhärenter (Teil-)Intelligenz. Im weiteren Verlauf beschäftigen sich die Studierenden mit geeigneten Entwurfs- und Spezifikationstechniken für die Entwicklung von ITS, primär dem Systems Engineering. Hierzu bietet die Vorlesung eine Einführung in die grundlegenden Methoden und Architekturen für ITS, die anhand aktueller Beispiele erläutert werden. Methoden für Analyse und Test von ITS auf Basis einer Systemspezifikation zeigen abschließend, dass eine ganzheitliche Vorgehensweise (das Systems Engineering) beim Entwurf der entscheidende Erfolgsfaktor für die technischen Systeme von morgen ist.

- Einführung in die Grundlagen von ITS
- Historie des Systems Engineerings
- Systemtheorie und Systemdenken
- Prozesse und Richtlinien des Systems Engineerings
- Hilfsmittel für die Planung und den Entwurf von ITS

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Ziel der Vorlesung ist die Vermittlung von Grundwissen des Systems Engineering und Intelligenten technischen Systemen (ITS). Die Studierenden sind in der Lage, Methoden des Sytems Engineering anzuwenden und auf Beispiele von ITS anzuwenden. Wesentliche ist dabei die Aneignung von Wissen zum Thema Systemtheorie und der Anwendung des Systemdenkens, auf technische und nicht technische Systeme. Hierfür werden Richtlinien und Werkzeuge für die Planung und Entwicklung von ITS vorgestellt und angewendet.
Nichtkognitive Kompetenzen:
• Gruppenarbeit
• Einsatz und Engagement</td></tr>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a)</td><td>Klausur oder mündliche Prüfung</td><td>90-120 min bzw. 40 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Roman Dumitrescu</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Systems Engineering:* **Lernmaterialien, Literaturangaben** • acatech (Hrsg.): Cyber-Physical Systems – Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion (acatech POSITION). Springer-Verlag, 2011 • Gausemeier, J.; Dumitrescu, R.; Steffen, D.; Czaja, A.; Wiederkehr, O.; Tschirner, C.: Systems Engineering in der industriellen Praxis. Paderborn, 2013

User Interface Design (BA Inf v4)

User Interface Design (BA CS v4)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2130	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5	1	en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05499 User Interface Design	V2 Ü3	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

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 „Analysis für Informatiker“ und „Lineare Algebra für Informatiker“ berücksichtigt.

4 **Inhalte:**

Inhalte der Lehrveranstaltung User Interface Design:
Software muss benutzbar sein, um ihren Zweck erfüllen zu können. Schlechte Bedienbarkeit bedeutet in der Regel Frustration, Verschwendung von Ressourcen wie Zeit, Aufmerksamkeit und Geld, sowie Ausgrenzung “nicht bedachter” Personengruppen. In dieser Veranstaltung erlernen wir die Planung, das Design, die Implementierung und Evaluierung von benutzbarer Software, insbesondere User Interfaces. Wir gehen dabei auf Benutzbarkeit (Usability) ein im Sinne von Erlernbarkeit, Barrierefreiheit (Accessibility), Inklusion, Ästhetik, Gestaltprinzipien, Design Patterns und Gamification. Wir arbeiten auf klassischen sowie mobilen Umgebungen. Wir bauen dabei auf physiologischen und psychologischen Grundlagen auf und gehen auch auf (un)ethischen Umgang mit diesen in der Softwareentwicklung ein.

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden

- besitzen die erforderlichen Fähigkeiten zum Entwerfen guter Benutzeroberflächen nach psychologischen Gestaltungsprinzipien, Designpatterns und Informationsarchitektur.
- haben Verständnis entwickelt für Nutzende, menschliches Verhalten und die Art und Weise, wie Menschen Benutzeroberflächen wahrnehmen, verarbeiten und mit ihnen interagieren.
- haben Fähigkeiten und Vokabular zum Verständnis und zur Bewertung der Accessibility aufgebaut.
- erkennen und vermeiden Deceptive Patterns und betrachten Nudges kritisch.
- haben Verständnis entwickelt für die Rolle der Ethik im Design und die Auswirkungen von Designentscheidungen auf Nutzende und Gesellschaft.
- sind mit den Begriffen vertraut, die den Einstieg in weitere Kurse und spätere Forschungen im Bereich der Mensch-Maschine-Interaktion erleichtert, auch mit der Schnittstelle von empirischer Softwareentwicklung und IT-Sicherheit.

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur oder mündliche Prüfung	90-120 min bzw. 30-45 min	100%

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		SL

Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Studienleistung bzw. qualifizierte Teilnahme konkret zu erbringen ist.

8 **Voraussetzungen für die Teilnahme an Prüfungen:**

Bestehen der Studienleistung

9 **Voraussetzungen für die Vergabe von Credits:**

Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).

11 **Verwendung des Moduls in anderen Studiengängen:**

Bachelorstudiengang Informatik v4 (IBA v4)

12	**Modulbeauftragte/r:** Prof. Dr. Yasemin Acar
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung User Interface Design:* **Methodische Umsetzung:** Die Vorlesung ist mit interaktiven Lernquizzes gestaltet; Übungsgruppen dienen der praktischen Anwendung von Gelerntem sowie Klärung von Fragen bei der Bearbeitung der Hausübungen. Die Hausübungen werden von Tutor*innen bewertet und sind, möglicherweise gemeinsam mit den Quizzes, Grundlage für die Studienleistung. **Lernmaterialien, Literaturangaben:** Vorlesungsaufzeichnungen, Vorlesungsfolien (auf Englisch) und Übungsblätter.

Ausgewählte Themen im Gebiet Softwaretechnik

Selected Topics in Software Engineering (BA)

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2117	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5-6	1	de / en

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**

2 **Wahlmöglichkeiten innerhalb des Moduls:**

Innerhalb des Moduls ist genau eine der enthaltenen Lehrveranstaltungen zu wählen.

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

In diesem Modul werden wechselnde Lehrveranstaltungen, beispielsweise von Gastdozenten angeboten. Die konkreten Inhalte sind den Lehrveranstaltungen zu entnehmen.

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden sind in der Lage für ein spezifisches Themenfeld aus dem Gebiet Softwaretechnik:

- Problemstellungen adäquat zu beschreiben
- Problemstellungen zu analysieren und Lösungsansätze zu erarbeiten
- Lösungen zu gegebenen Problemstellungen zu erarbeiten und umzusetzen
- Lösungen zu gegebenen Problemstellungen zu bewerten
- Verschiedene Lösungen zu gegebenen Problemstellungen zu vergleichen, zu bewerten und eine adäquate Lösung auszuwählen

6 **Prüfungsleistung:**

☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
	Klausur, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

keine

8	**Voraussetzungen für die Teilnahme an Prüfungen:** Bestehen der Studienleistung
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Studiengangsbeauftragter Informatik
13	**Sonstige Hinweise:**

3.5 Vertiefung

Dieses Gebiet erlaubt Studierenden eine Vertiefung in einem der vier Gebiete der Informatik (Algorithmen und Komplexität, Computersysteme, Daten und Wissen, Softwaretechnik). Es kann ein beliebiges Modul aus einem dieser Gebiete im 2. Studienabschnitt gewählt werden.

4 Weiteres

Schlüsselqualifikation

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2510	150	5	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	beliebiges Semester	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	Proseminar	PS2	15	105	P	15
b)	Mentoring	Treffen in Kleingruppen	15	15	P	20

2 **Wahlmöglichkeiten innerhalb des Moduls:**

Für das Proseminar können alle Proseminare aus dem Angebot des Bachelorstudiengangs Informatik gewählt werden.

3 **Teilnahmevoraussetzungen:**

Teilnahmevoraussetzungen der Lehrveranstaltung Proseminar:
Empfohlene Vorkenntnisse
Je nach gewähltem Thema.

4	**Inhalte:** Das Modul besteht aus zwei Teilen: dem Proseminar und dem Mentoring. *Inhalte der Lehrveranstaltung 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. Die vermittelte Inhaltskompetenz betreffs der fachlichen Ausrichtung des Proseminars ist abhängig vom jeweiligen Thema der Veranstaltung. Unabhängig davon werden den Studierenden im Proseminar Fakten zur Erstellung und dem Ablauf von Präsentationen (Medienkompetenz), sowie dem Umgang mit Literatur, sowie zur Anfertigung schriftlicher Ausarbeitungen vermittelt. *Inhalte der Lehrveranstaltung 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 Faches 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.
5	**Lernergebnisse und Kompetenzen:** Im Proseminar wird neben dem inhaltlichen Aspekt vor allem das Aufbereiten eines Themas und seine Präsentation eingeübt. Die Studierenden lernen in der praktischen Durchführung das Erarbeiten eines Themas, das Treffen von Auswahlen, das Halten von Vorträgen, den Umgang mit Fragen und Diskussionsbeiträgen, sowie das Anfertigen von größeren schriftlichen Texten. Die erarbeiteten Kompetenzen im Proseminar bereiten das Bewältigen ähnlicher Situationen später im Studium (Seminar, Projektgruppe, Abschlussarbeit) und im Beruf (Präsentationen, Berichte) vor. **Nichtkognitive Kompetenzen:** • Einsatz und Engagement • Haltung und Einstellung • Lernkompetenz • Lernmotivation • Medienkompetenz • Motivationale und volitionale Fähigkeiten • Schreib- und Lesekompetenz (wissenschaftlich) • Selbststeuerungskompetenz

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table>
<tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr>
<tr><td>a) - b)</td><td>Seminarvortrag und schriftliche Ausarbeitung</td><td>45-60 min</td><td>100%</td></tr>
</table>
Vom jeweiligen Lehrenden werden Art und Dauer der Prüfungsleistung spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Prüfungsleistung bestanden und das Mentoring absolviert worden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der doppelten Anzahl seiner Credits gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Studiengangsbeauftragter Informatik</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Proseminar:
Methodische Umsetzung
Referate mit schriftlicher Ausarbeitung und Vortrag.
Lernmaterialien, Literaturangaben
Je nach gewähltem Thema.

Hinweise der Lehrveranstaltung Mentoring:
Es finden während des gesamten Bachelorstudiums je nach Bedarf etwa zweimal im Semester Treffen statt, in Kleingruppen oder individuell.</td></tr>
</table>

Studium Generale – Bachelor			
General Studies – Bachelor			
Modulnummer: null	**Workload (h):** 210	**Leistungspunkte:** 7	**Turnus:** module.frequency.de
	Studiensemester: Beliebig	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	Studium Generale – Bachelor	V3 Ü2	75	135	P	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

Beliebige Veranstaltungen außerhalb der Informatik können gewählt werden.

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

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.

Inhalte der Lehrveranstaltung Studium Generale – Bachelor:
Abhängig von den gewählten Veranstaltungen

5 **Lernergebnisse und Kompetenzen:**

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.
Nichtkognitive Kompetenzen:

- Einsatz und Engagement
- Kooperationskompetenz
- Medienkompetenz
- Schreib- und Lesekompetenz (wissenschaftlich)

<table>
<tr><td>6</td><td>Prüfungsleistung:
☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)
<table><tr><th>zu</th><th>Prüfungsform</th><th>Dauer bzw. Umfang</th><th>Gewichtung für die Modulnote</th></tr><tr><td>a)</td><td>Prüfung im Studium Generale</td><td></td><td>100%</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die Prüfungsleistung konkret zu erbringen ist.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
<table><tr><th>zu</th><th>Form</th><th>Dauer bzw. Umfang</th><th>SL / QT</th></tr><tr><td>a)</td><td>Qualifizierte Teilnahme im Rahmen des Studium Generale</td><td></td><td>QT</td></tr></table>Vom jeweiligen Lehrenden wird spätestens in den ersten drei Wochen der Vorlesungszeit bekannt gegeben, wie die qualifizierte Teilnahme konkret zu erbringen ist.</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Credits gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v4 (IBA v4)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Studiengangsbeauftragter Informatik</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:</td></tr>
</table>

Bachelor-Abschlussarbeit			
Bachelor Thesis			
Modulnummer: A.079.2910	**Workload (h):** 450	**Leistungspunkte:** 15	**Turnus:** module.frequency.de
	Studiensemester: 6	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	Bachelor-Abschlussarbeit – Arbeitsplan		15	75	P	1
b)	Bachelor-Abschlussarbeit		15	345	P	1

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

Erfolgreicher Abschluss aller Module des ersten Studienabschnitts sowie des Moduls Schlüsselqualifikation.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Bachelor-Abschlussarbeit – 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.

Inhalte der Lehrveranstaltung Bachelor-Abschlussarbeit:
Die 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.

5	**Lernergebnisse und Kompetenzen:** 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. **Nichtkognitive Kompetenzen:** • Einsatz und Engagement • Lernmotivation • Motivationale und volitionale Fähigkeiten • Schreib- und Lesekompetenz (wissenschaftlich) • Selbststeuerungskompetenz
6	**Prüfungsleistung:** ☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a) - b)	Abschlussarbeit		100%

7	**Studienleistung, qualifizierte Teilnahme:** keine
8	**Voraussetzungen für die Teilnahme an Prüfungen:**
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit 48 Credits gewichtet.
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v4 (IBA v4)
12	**Modulbeauftragte/r:** Studiengangsbeauftragter Informatik

13	**Sonstige Hinweise:** 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. *Hinweise der Lehrveranstaltung Bachelor-Abschlussarbeit – Arbeitsplan:* **Methodische Umsetzung** Direkte Absprache mit Betreuer. **Lernmaterialien, Literaturangaben** Je nach gewähltem Thema. *Hinweise der Lehrveranstaltung Bachelor-Abschlussarbeit:* **Methodische Umsetzung** Selbständiges Arbeiten unterstützt durch individuelle Betreuung **Lernmaterialien, Literaturangaben** Je nach gewähltem Thema.

5 Module im Wintersemester

- M.079.2232 3D Modeling (BA Inf v4) 55
- M.105.9710 Analysis für Informatiker 45
- M.079.2420 Angewandte Statistik mit R (BA Inf v4) 101
- M.079.2217 Ausgewählte Themen im Gebiet Algorithmen und Komplexität 81
- M.079.2317 Ausgewählte Themen im Gebiet Computersysteme 99
- M.079.2417 Ausgewählte Themen im Gebiet Daten und Wissen 119
- M.079.2117 Ausgewählte Themen im Gebiet Softwaretechnik 140
- A.079.2910 Bachelor-Abschlussarbeit 148
- M.079.1212 Berechenbarkeit und Komplexität 29
- M.079.2412 Databases and Information Systems (BA Inf v4) 104
- M.079.2210 Einführung in Kryptographie (BA Inf v4) 62
- M.079.2211 Grundlegende Algorithmen (BA Inf v4) 68
- M.079.1313 IT-Sicherheit 42
- M.079.2430 Knowledge Base Engineering (BA Inf v4) 116
- M.079.2212 Komplexitätstheorie (BA Inf v4) 71
- M.079.2114 Modellbasierte Softwareentwicklung (BA Inf v4) 125
- M.079.1210 Modellierung 22
- M.079.1111 Programmiersprachen 7
- M.079.1110 Programmierung 4
- M.079.2230 Quantum Networks (BA Inf v4) 74
- M.079.1311 Rechnerarchitektur 35
- M.079.2510 Schlüsselqualifikation 143
- M.079.2115 Secure Software Engineering (BA CS v4) 128
- M.079.1113 Softwaretechnikpraktikum 17
- M.105.9730 Stochastik für Informatiker 51
- Studium Generale – Bachelor 146
- M.079.2130 User Interface Design (BA Inf v4) 137
- M.079.2313 Verteilte Systeme (BA Inf v4) 96

6 Module im Sommersemester

- M.079.2231 3D Rendering (BA Inf v4) 59
- M.079.2131 Applied User Experience Design (BA Inf v4) 121
- M.079.2217 Ausgewählte Themen im Gebiet Algorithmen und Komplexität 81
- M.079.2317 Ausgewählte Themen im Gebiet Computersysteme 99
- M.079.2417 Ausgewählte Themen im Gebiet Daten und Wissen 119
- M.079.2117 Ausgewählte Themen im Gebiet Softwaretechnik 140
- A.079.2910 Bachelor-Abschlussarbeit 148
- M.079.1114 Datenbanksysteme 10
- M.079.1211 Datenstrukturen und Algorithmen 25
- M.079.1310 Digitaltechnik 32
- M.079.2318 Einführung in Hochleistungsrechnen (BA Inf v4) 83
- M.048.11113 Einführung in Planungs- und Entscheidungsfindungsalgorithmen 87
- M.079.2311 Eingebettete Systeme (BA Inf v4) 90
- M.079.2331 Embedded Machine Learning (BA Inf v4) 107
- M.079.2414 Foundations of the Semantic Web (BA Inf v4) 110
- M.079.1115 Gestaltung von Nutzungsschnittstellen 20
- M.079.2215 Grundlagen der Algorithmische Geometrie (BA Inf v4) 65
- M.079.2431 Introduction to Knowledge Representation and Reasoning (BA Inf v4) 113
- M.105.9720 Lineare Algebra für Informatiker 48
- M.079.2510 Schlüsselqualifikation 143
- M.079.1112 Software Engineering 13
- Studium Generale – Bachelor 146
- M.079.2118 Systematische Entwicklung von VR/AR Anwendungen (BA Inf v4) 130
- M.079.2119 Systems Engineering (BA Inf v4) 134
- M.079.1312 Systemsoftware und systemnahe Programmierung 38
- M.079.2214 Verteilte Algorithmen und Datenstrukturen (BA Inf v4) 77

7 Modules in English

- M.079.2131 Applied User Experience Design (BA CS v4) 121
- M.079.2412 Databases and Information Systems (BA CS v4) 104
- M.079.2313 Distributed Systems (BA CS v4) 96
- M.079.2331 Embedded Machine Learning (BA CS v4) 107
- M.079.2414 Foundations of the Semantic Web (BA CS v4) 110
- M.079.2211 Fundamental Algorithms (BA CS v4) 68
- M.079.2210 Introduction to Cryptography (BA CS v4) 62
- M.079.2431 Introduction to Knowledge Representation and Reasoning (BA CS v4) 113
- M.079.2430 Knowledge Base Engineering (BA CS v4) 116
- M.079.2230 Quantum Networks (BA CS v4) 74
- M.079.2115 Secure Software Engineering (BA CS v4) 128
- M.079.2217 Selected Topics in Algorithms and Complexity 81
- M.079.2317 Selected Topics in Computer Systems (BA) 99
- M.079.2417 Selected Topics in Data and Knowledge (BA) 119
- M.079.2117 Selected Topics in Software Engineering (BA) 140
- M.079.2130 User Interface Design (BA CS v4) 137

Erzeugt am 30. September 2026 um 16:34.