UNIVERSITÄT PADERBORN

FAKULTÄT FÜR ELEKTROTECHNIK, INFORMATIK UND MATHEMATIK

INSTITUT FÜR INFORMATIK

MODULHANDBUCH FÜR DEN BACHELORSTUDIENGANG INFORMATIK V5 (IBA V5)

STAND: 30. SEPTEMBER 2026

Inhaltsverzeichnis

1 Präambel und Hinweise **3**

2 Basisstudium **4**
2.1 1. Semester 4
2.2 2. Semester 16
2.3 3. Semester 31
2.4 4. Semester 47
2.5 5. Semester 61

3 Vertiefungsstudium **67**
3.1 Algorithmen und Komplexität 67
3.2 Computersysteme 93
3.3 Daten und Wissen 110
3.4 Softwaretechnik 130

4 Module im Wintersemester **155**

5 Module im Sommersemester **156**

6 Modules in English **157**

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.

2 Basisstudium

Im Basisstudium werden die wesentlichen Grundkenntnisse und Fähigkeiten vermittelt, die alle Absolvent*innen der Informatik kennen und beherrschen sollten. Das Basisstudium besteht ausschließlich aus Pflichtveranstaltungen.

2.1 1. Semester

Einführung in die Informatik

Introduction to Computer Science

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.1521	90	3	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	1	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05101 Einführung in die Informatik	V2	30	60	P	300

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4	**Inhalte:** *Inhalte der Lehrveranstaltung Einführung in die Informatik:* Die Informatik ist eine junge Wissenschaft, die sich in den letzten Jahrzehnten rasant entwickelt hat. Die Grundlagen, Methoden und Produkte der Informatik sind zunehmend wichtig für andere Wissenschaften und für viele gesellschaftlich relevante Entwicklungen. Diese Veranstaltung stellt zunächst die Entwicklung der Informatik und ihre Wechselwirkungen mit anderen Wissenschaftsgebieten und der Gesellschaft vor und dient somit der Identitätsstiftung für angehende Informatiker:innen. Im Anschluss werden Strukturen an der Universität Paderborn, insbesondere solche zur Unterstützung des individuellen Studienverlaufs, eingeführt und Berufsbilder für Informatiker:innen aufgezeigt. Die Veranstaltung umfasst folgende Inhalte: • Entwicklung der Informatik als Wissenschaft • Bezüge der Informatik zu anderen Wissenschaften und zur Gesellschaft • Teilgebiete der Informatik, Aufbau des Informatikstudiums an der Universität Paderborn • Strukturen an der Universität Paderborn, Unterstützung während des Studiums • Soft skills: Zeitmanagement, Lernen lernen, Anregung zur Selbstreflexion über den persönlichen Studienverlauf • Vorstellung von Fachgebieten der Informatik in Paderborn • Berufsbilder für Informatiker:innen
5	**Lernergebnisse und Kompetenzen:** Die Studierenden können • Meilensteine in der Entwicklung der Informatik benennen, • die Fachsystematik der Informatik erklären, • die Informatik in den Kanon der Wissenschaften einordnen, • ihren studentischen Alltag organisieren, • ihren persönlichen Studienverlauf reflektieren, • ihren weiteren Studienverlauf planen.
6	**Prüfungsleistung:** keine
7	**Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Testat	ca. 5 DIN-A4-Seiten	QT

8	**Voraussetzungen für die Teilnahme an Prüfungen:**
9	**Voraussetzungen für die Vergabe von Credits:** Die Vergabe von Credits erfolgt, wenn die qualifizierte Teilnahme erbracht wurde.

10	**Gewichtung für Gesamtnote:** Das Modul ist unbenotet.
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Prof. Dr.-Ing. Juraj Somorovsky
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Einführung in die Informatik:* **Methodische Umsetzung** Die Veranstaltung besteht aus einem ersten Teil mit Präsentationen im Vorlesungsstil und anschließender Diskussion, sowie einem zweiten Teil im Stil einer Ringvorlesung, in der unterschiedliche Dozent:innen vortragen. **Lernmaterialien, Literaturangaben** • Präsentationsfolien • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Programmierung 1			
Programming 1			
Modulnummer: M.079.1121	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 1	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05100 Programmierung 1	V3 Ü4	105	135	P	450/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Programmierung 1:
Die Veranstaltung richtet sich an Studierende mit wenigen oder keinen Vorkenntnissen in der Softwareentwicklung. Sie führt die Studierenden an diese Tätigkeit heran, indem sie auf systematische und wissenschaftlich fundierte Art und Weise die grundlegenden Bausteine der imperativen und objektorientierten Programmierung in Python vermittelt und deren Anwendung praktisch einübt.
Die Veranstaltung umfasst folgende Inhalte:

- Grundlegende Programmierbausteine (Variablen, Operatoren und Ausdrücke, Zuweisungen, Verzweigungen, Schleifen, Ein- und Ausgabe)
- Elementare Datentypen und Typverträglichkeit (Integer, String, Float, Boolean)
- Funktionen (Definition, Aufruf, Parameter, Rückgabewerte)
- Automatische Tests (Entwurf, Implementierung, Ausführung)
- Datentypen (Arrays, Dictionaries)
- Rekursion (rekursive Problemreduktion, Abbruchbedingung, Zusammenhang zur Iteration)
- Klassen (Klassen und Objekte, Attribute und Methoden, Überladung, Sichtbarkeit, Delegation, Gleichheit und Identität, statische und nicht-statische Attribute)
- Ausnahmen (Definition, Behandlung)
- Dateizugriff und Betriebssystemaufrufe (Lesen und Schreiben, sequentiell und Direktzugriff)

5 Lernergebnisse und Kompetenzen:

Nach Abschluss dieses Moduls können Studierende

- Syntax und Semantik der wesentlichen Sprachkonstrukte einer objektorientierten Sprache erläutern
- Grundkonzepte der imperativen und objektorientierten Programmierung erläutern, einfache algorithmische Lösungsmuster nachvollziehen und diese passend für gegebene Problemstellungen auswählen
- einfache Algorithmen und Objektstrukturen zur Problemlösung entwerfen, implementieren und testen
- für einfache Anwendungsprobleme eine geeignete programmtechnische Lösung auswählen
- grundlegende Programmentwurfsprinzipien und -methoden anwenden
- der Anwendungsdomäne entsprechende einfache Objektstrukturen entwerfen
- die grundlegenden Werkzeuge der Softwareentwicklung effektiv einsetzen
- Lösungsansätze für Programmierprobleme gemeinsam in Kleingruppen entwickeln
- Lösungsansätze für Programmierprobleme bewerten
- eigene analytische und konzeptionelle Fähigkeiten einschätzen

6 Prüfungsleistung:

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

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

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben oder Kurzklausur		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 Modulabschlussprüfung bestanden ist.

10 Gewichtung für Gesamtnote:

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

11 Verwendung des Moduls in anderen Studiengängen:

Bachelorstudiengang Informatik v5 (IBA v5)

12 Modulbeauftragte/r:

Prof. Dr. Stefan Böttcher

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Programmierung 1:* **Methodische Umsetzung** Die Inhalte werden im Vorlesungsteil an typischen Beispielen eingeführt, in den praktischen Übungen unter Anleitung erprobt und in Übungsaufgaben in eigenständiger Bearbeitung der Studierenden vertieft. Dabei wird die Anwendung der wichtigsten Softwareentwicklungswerkzeuge wie Editor, Interpreter und Debugger eingeübt. **Literaturangaben** • Mark Lutz: Learning Python, 5th ed., O'Reilly • Thomas Theis: Einstieg in Python, Rheinwerk Computing • Weitere Literatur wird in der Veranstaltung bekanntgegeben

Modellierung			
Modelling			
Modulnummer: M.079.1221	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 1	**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	450/25

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

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden können
• grundlegende Literatur im Bereich der Modellierung verstehen,
• für ein gegebenes Problem eine geeignete Modellierungstechnik auswählen und das Problem mit dieser Technik beschreiben
Der erfolgreiche Abschluss dieses Moduls ermächtigt Studierende, erlernte Modellierungstechniken anzuwenden, grundsätzliche Techniken zu erweitern und verfeinern, um somit neuartige Probleme zu modellieren. Sie werden darüber hinaus ermächtigt, diese im Techniken im Rahmen verschiedener Anwendungen zu implementieren, analysieren und bewerten.</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>120-180 min</td><td>100%</td></tr>
</table></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 Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5), Bachelorstudiengang Nachhaltiger Maschinenbau V1, Bachelorstudiengang Nachhaltiger Maschinenbau V1b</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Christian Scheideler, Prof. Dr.-Ing. Anni-Yasmin Turhan</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Modellierung:* **Methodische Umsetzung** In der Vorlesung werden • die grundlegenden Begriffe eingeführt, • mathematische Konzepte formuliert, analysiert und in Teilen bewiesen • die theoretischen Inhalte anhand einfacher Beispiele veranschaulicht In den Übungen werden • die erlernten Konzepte umfassend auf Fragestellungen verschiedener Komplexität angewendet • weiter führende Beispiele betrachtet, um das Erlernte zu erweitern bzw. um weitere Perspektiven zu ergänzen Die Studienleistung • dient der besseren Vorbereitung auf die Abschlussprüfung • motiviert Studierende, kontinuierlich mitzuarbeiten • ermöglicht regelmäßige Rückmeldungen über den eigenen Lernfortschritt **Lernmaterialien, Literaturangaben** • Uwe Kastens, Hans Kleine Büning: Modellierung • Angelika Steger: Diskrete Strukturen • Foliensatz der Vorlesung; Übungsblätter

Lineare Algebra für Informatik			
Linear Algebra for Computer Science			
Modulnummer: M.105.9610	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 1	**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.105.96100 Lineare Algebra für Informatik	V4 Ü2	90	150	P	450/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Lineare Algebra für Informatik:
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 hinaus gehend 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
   - Mengen und Abbildungen
   - Vollständige Induktion und Rekursion, Kombinatorik
   - Elementare Zahlentheorie
   - Reelle und komplexe Zahlen, Körper
2. Vektorräume
3. lineare Abbildungen
4. Basis
5. Dimension
6. Matrizen
7. lineare Gleichungssysteme
8. Determinanten

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.

Der erfolgreiche Abschluss dieses Moduls ermächtigt Studierende, mathematische Fragestellungen aus dem Bereich der linearen Algebra zu verstehen und interpretieren sowie diese im Rahmen verschiedener Anwendungen zu implementieren, analysieren und bewerten.

6 Prüfungsleistung:

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

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

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben		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 Modulabschlussprüfung bestanden ist.

10 Gewichtung für Gesamtnote:

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

11 Verwendung des Moduls in anderen Studiengängen:

Bachelorstudiengang Informatik v5 (IBA v5)

12	**Modulbeauftragte/r:** Prof. Dr. Jürgen Klüners
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Lineare Algebra für Informatik:* **Methodische Umsetzung** In der Vorlesung werden • die grundlegenden Begriffe eingeführt, • mathematische Konzepte formuliert, analysiert und in Teilen bewiesen • die theoretischen Inhalte anhand einfacher Beispiele veranschaulicht In den Übungen werden • die erlernten Konzepte umfassend auf Fragestellungen verschiedener Komplexität angewendet • weiter führende Beispiele betrachtet, um das Erlernte zu erweitern bzw. um weitere Perspektiven zu ergänzen Die Studienleistung • dient der besseren Vorbereitung auf die Abschlussprüfung • motiviert Studierende, kontinuierlich mitzuarbeiten • ermöglicht regelmäßige Rückmeldungen über den eigenen Lernfortschritt

2.2 2. Semester

Programmierung 2

Programming 2

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.1122	240	8	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	2	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05201 Programmierung 2	V3 Ü4	105	135	P	400/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Programmierung 2:
Empfohlene Vorkenntnisse
Kenntnisse aus dem Modul *Programmierung 1*.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Programmierung 2:
Die Veranstaltung setzt auf den Kenntnissen der Lehrveranstaltung *Programmierung I* auf und erweitert und vertieft diese. Dabei liegt der Schwerpunkt auf objektorientierter Programmierung sowie der Konzeption und Implementierung von grundlegenden Algorithmen und Datenstrukturen wie Such- und Sortierverfahren, Bäumen und Graphen und ihrer Traversierung sowie auf der korrekten Implementierung nebenläufiger Programme.
Die Veranstaltung umfasst folgende Inhalte:

- Vererbung (Ober- und Unterklassen, Überschreiben und Überdecken, abstrakte Methoden und Klassen, multiple Vererbung)
- Bibliotheken (Einbindung, Anwendung)
- Array-basierte Such- und Sortieralgorithmen (binäre Suche, Rank, Bubble-, Index- und Radix-Sort)
- Dynamische Datenstrukturen (lineare Liste, Ringliste, Binär- und allgemeine Bäume, Tiefensuche, Ausdrucksbäume und ihre Auswertung)
- Elementare Algorithmen auf Graphen (Implementierung von Graphen durch Arrays, Tiefen- und Breitensuche, Zyklensuche)
- Backtracking-Algorithmen (z.B. 8-Damen-Problem)
- Nebenläufige Programmierung (Threads, Generatoren, Stoppen und Aufwecken von Prozessen, kritische Abschnitte, Lese-/Schreibsperren, Warteschlangen und beschränkte Puffer, 5-Philosophen-Problem)

5 **Lernergebnisse und Kompetenzen:**

Nach Abschluss dieses Moduls können Studierende

- fortgeschrittene Konzepte objektorientierter Programmiersprachen erklären und anwenden
- die grundlegenden Such- und Sortieralgorithmen erklären, implementieren und anwenden
- dynamische Datenstrukturen verstehen, sinnvoll einsetzen und implementieren
- Datenstrukturen und Algorithmen aus Softwarebibliotheken finden und nutzen
- selbständig dynamische Datenstrukturen entwerfen und implementieren
- selbständig vollständige Programme kleinen und mittleren Umfangs erstellen
- einfache nebenläufige Programme erstellen
- typische Probleme bei der nebenläufigen Programmierung identifizieren und lösen
- Lösungsansätze für Programmierprobleme gemeinsam in Kleingruppen entwickeln
- Lösungsansätze für Programmierprobleme und deren Aufwand bewerten
- eigene analytische und konzeptionelle Fähigkeiten einschätzen

6 **Prüfungsleistung:**

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

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

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben oder Kurzklausur		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 Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Stefan Böttcher

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Programmierung 2:
Methodische Umsetzung
Die Inhalte werden im Vorlesungsteil an typischen Beispielen eingeführt, in den praktischen Übungen unter Anleitung erprobt und in Übungsaufgaben in eigenständiger Bearbeitung der Studierenden vertieft. Dabei wird die Anwendung der wichtigsten Softwareentwicklungswerkzeuge wie Editor, Interpreter und Debugger eingeübt.
Literaturangaben

- Python-Tutorials von `https://www.pdfdrive.com/python-tutorial-books.html`
- Johannes Ernesti, Peter Kaiser: Python 3 - Das umfassende Handbuch, Rheinwerk Computing
- Weitere Literatur wird in der Veranstaltung bekanntgegeben

Datenbanken			
Databases			
Modulnummer: M.079.1421	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 2	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05203 Datenbanken	V3 Ü2	75	105	P	400/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Datenbanken:
Empfohlene Vorkenntnisse:
Kenntnisse in der Programmierung werden in dem Umfang vorausgesetzt, wie sie im Modul *Programmierung 1* gelehrt werden. Elementare Kenntnisse der Logik und der Modellierung aus dem Modul *Modellierung* werden empfohlen.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Datenbanken:* Datenbanken spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des Wissens 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. Durch die Nutzung zunehmend größerer Datenbestände (Big Data) entstehen neuartige Anforderungen (Umgang mit Volume, Variety, Velocity), die neue Techniken in der Datenbankwelt erfordern. So werden verteilte Datenbanken immer wichtiger, und der Umgang mit Netzwerkunterbrechungen erfordert geschickte, je nach Anwendung unterschiedliche Kompromisse zwischen Erreichbarkeit, Aktualität und Konsistenz verteilter Daten. NoSQL-Datenbanken sind für diese speziellen Bedürfnisse konzipiert und setzen – im Gegensatz zu den traditionellen relationalen Datenbanksystemen – den Fokus auf eine deutlich flexiblere Datenorganisation. Dieses Modul erschließt die Grundlagen für verschiedenartige Datenbanksysteme, die in nahezu allen Unternehmen in der Praxis eingesetzt werden. Die Inhalte umfassen im Einzelnen: • Relationales Datenmodell und relationale Algebra • SQL – Datendefinitonssprache, Datenmanipulationssprache und Anfragesprache; Stored Procedures; Assertions • Zugriffskontrolle und Sichten – Views in SQL; Rechtevergabe in SQL • Physische Datenorganisation und Indizes • Anfrageoptimierung • Datenintegrität • Funktionale Abhängigkeiten und Datenbankschemaentwurf – Normalformen; Transformationseigenschaften; Integritätsbedingungen (Schlüssel und Fremdschlüssel); Trigger in SQL • Transaktionen – Synchronisation; Recovery; Atomic Commit Protokolle • NoSQL-Datenbanken – Graph-Datenbanken; Dokumenten-orientierte Datenbanken; Key-Value Stores; Column Stores; Frameworks zur Auswertung verteilter Daten/Streamprocessing (Map/Reduce, Apache Spark, Apache Flink) • Hauptspeicher-Datenbanken • Eingebettetes SQL – SQL-Einbettung in Java bzw. Python; SQL Injection und Prepared Statements • Datenschutz in Datenbanken – Herausforderungen und Grenzen

5 **Lernergebnisse und Kompetenzen:**

Nach Abschluss dieses Moduls können Studierende

- Operatoren der relationalen Anfragesprachen benennen und deren Bedeutungen erklären
- Anfragen in relationaler Algebra, SQL und Cypher korrekt interpretieren und formulieren und an existierende relationale Datenbanken bzw. Graph-Datenbanken stellen.
- Programme schreiben, die Datenbestände aus Datenbanken lesen oder verändern
- ein Datenbankschema möglichst redundanzfrei entwerfen und darauf aufbauend eine Datenbank definieren und aufbauen. Weiterhin können die Studierenden die Risiken eines schlecht entworfenen Datenbankschemas bewerten.
- verteilte Datenanalyse mit Map/Reduce, Apache Spark und Apache Flink durchführen.
- den Programmieraufwand für Datenbankanfragen und Datenbankprogrammierung einschätzen sowie Anfragen manuell optimieren.
- die Folgen einer Datenbankschema-Änderung erkennen und abschätzen
- den Aufwand und Nutzen von Synchronisation und Recovery verstehen und die Techniken und Verfahren in diesen Bereichen praktisch einsetzen.
- die Eignung und Grenzen des relationalen Datenmodells bewerten und einschätzen und mit alternativen Datenmodellen vergleichen
- die erworbenen Kompetenzen und Fertigkeiten auf andere Datenquellen oder andere Datenbanksysteme übertragen
- mit anderen Studierenden in Kleingruppen kooperieren, um gemeinsam eine geeignete Lösung für eine gegebene Aufgabe im Bereich der vermittelten Inhalte zu erstellen

6 **Prüfungsleistung:**

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

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

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben		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 Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v5 (IBA v5), Bachelorstudiengang Nachhaltiger Maschinenbau V1, Bachelorstudiengang Nachhaltiger Maschinenbau V1b
12	**Modulbeauftragte/r:** Dr. Rita Hartel, Prof. Dr. Stefan Böttcher
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Datenbanken:* **Methodische Umsetzung** Die Grundlagen und Konzepte von Datenbanksystemen werden in einer Präsentation im Rahmen einer Vorlesung eingeführt und anschließend in Präsenzübungen in Kleingruppen sowie in Übungsaufgaben vertieft und durch praktische Übungen ergänzt. **Lernmaterialien, Literaturangaben** • Lehrbuch: Kemper, Eickler: Datenbanksysteme. Eine Einführung. Oldenbourg-Verlag, neueste Ausgabe. • Lehrbuch: Garcia-Molina, Ullman, Widom: Database Systems: The Complete Book, Prentice Hall, neueste Ausgabe. • Lehrbuch: Heuer, Saake: Datenbanksysteme – Konzepte und Sprachen. Mitp-Verlag, neueste Ausgabe. • Lehrbuch: Sadalage, Fowler: NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Addison-Wesley Professional, neueste Ausgabe.

Digitaltechnik			
Digital Design			
Modulnummer: M.079.1321	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 2	**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.079.05200 Digitaltechnik	V2 Ü2	60	120	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Digitaltechnik:
Empfohlene Vorkenntnisse
Kenntnisse aus dem Modul *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 und mit modernen Entwurfswerkzeugen umgesetzt.
Die Veranstaltung umfasst folgende Inhalte:

- Boolesche Algebra
- Gatter und Schaltnetze
- Logikoptimierung (Optimierung zweistufiger Logik nach Quine/McCluskey)
- Automaten und Schaltwerke (festverdrahtet, mikroprogrammierbar)
- Darstellung von Information und fehlerkorrigierende Codes
- 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 können
• den Entwurfsablauf in der Digitaltechnik von der Spezifikation bis zur technischen Realisierung beschreiben,
• die zugrunde liegenden mathematischen Modelle aus der Booleschen Algebra und der Automatentheorie anwenden,
• digitale Schaltungsentwürfe im Hinblick auf vorgegebene Entwurfsziele analysieren,
• einfache digitale Systeme selbständig konzipieren, sowie
• einfache digitale Systeme mit den entsprechenden Entwurfswerkzeugen technisch realisieren.</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></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 Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5)</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** Die Lehrveranstaltung besteht aus einer Vorlesung und Rechen- sowie praktischen Übungen. Die Vorlesung erfolgt mit Beamer und Tafelanschrieb. In den Rechenübungen werden Aufgaben ausgegeben und danach werden im Rahmen von Präsenzübungen in Kleingruppen die Lösungen durch die Übungsteilnehmer vorgestellt und diskutiert. In den praktischen Übungen wird ein Tutorial zum Schaltungsentwurf mit der Hardwarebeschreibungssprache VHDL durchgeführt und dann Aufgaben ausgegeben, die von den Übungsteilnehmern in Kleingruppen als Entwurfsbeispiele mit FPGA-Technologie umgesetzt werden. **Lernmaterialien, Literaturangaben** • Vorlesungsfolien, Aufgabenblätter für Rechenübungen • Tutorial, Aufgabenblätter für Entwurfsbeispiele und technische Dokumentationen für die praktischen Übungen • J. F. Wakerly, „Digital Design,“ 4th Edition, Upper Saddle River, NJ: Pearson / Prentice Hall, 2007 • Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien werden in der Veranstaltung bekannt gegeben.

Analysis für Informatik			
Calculus for Computer Science			
Modulnummer: M.105.9620	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 2	**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.105.96200 Analysis für Informatik	V4 Ü2	90	150	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Analysis für Informatik:
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.

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)

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).

Der erfolgreiche Abschluss dieses Moduls ermächtigt Studierende, mathematische Fragestellungen aus dem Bereich Analysis zu verstehen und interpretieren sowie diese im Rahmen verschiedener Anwendungen zu implementieren, analysieren und bewerten.

6 **Prüfungsleistung:**

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

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

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben		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 Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Jürgen Klüners

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Analysis für Informatik:* **Methodische Umsetzung** In der Vorlesung werden • die grundlegenden Begriffe eingeführt, • mathematische Konzepte formuliert, analysiert und in Teilen bewiesen • die theoretischen Inhalte anhand einfacher Beispiele veranschaulicht In den Übungen werden • die erlernten Konzepte umfassend auf Fragestellungen verschiedener Komplexität angewendet • weiter führende Beispiele betrachtet, um das Erlernte zu erweitern bzw. um weitere Perspektiven zu ergänzen Die Studienleistung • dient der besseren Vorbereitung auf die Abschlussprüfung • motiviert Studierende, kontinuierlich mitzuarbeiten • ermöglicht regelmäßige Rückmeldungen über den eigenen Lernfortschritt

Studium Generale – Bachelor			
General Studies – Bachelor			
Modulnummer: null	**Workload (h):** 360	**Leistungspunkte:** 12	**Turnus:** module.frequency.de
	Studiensemester: 2-6	**Dauer (in Sem.):** 5	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	2024.2005 Studium Generale – Bachelor	diverse	180	180	P	100/25

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 im Umfang von maximal 12 LP muss gewählt werden.

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.
Die Studierenden

- erwerben Wissen und Kompetenzen, die nicht fachspezifisch sind, aber für das angestrebte Berufsziel bedeutsam sein können wie beispielsweise spezielle Kenntnisse bei Fremdsprachen, in ingenieur-, natur-, kultur- oder wirtschaftswissenschaftlichen Gebieten,
- analysieren unterschiedlichste Fragestellungen in den entsprechenden Gebieten,
- ordnen das fachspezifische Wissen in einen interdisziplinären Zusammenhang ein,
- stellen Zusammenhänge zum Informatik-Studium her,
- erweitern ihre Schlüsselkompetenzen und ggf. Fremdsprachenkompetenzen, wodurch die Persönlichkeitsbildung unterstützt wird, auch in interkultureller Hinsicht.

6 **Prüfungsleistung:**

keine

<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>Qualifizierte Teilnahme im Rahmen des Studium Generale</td><td></td><td>QT</td></tr>
</table></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 qualifizierte Teilnahme erbracht wurde.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul ist unbenotet.</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Juraj Somorovsky</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:</td></tr>
</table>

2.3 3. Semester

Software Engineering

Software Engineering

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.1123	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	3	1	de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05300 Software Engineering	V2 Ü2	60	120	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Software Engineering:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Programmierung 1*, *Programmierung 2* und *Modellierung* sind hilfreich.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Software Engineering:* In der Veranstaltung werden die Grundlagen der systematischen und ingenieursmäßigen Softwareentwicklung vermittelt. Im Fokus steht dabei eine durchgängige modellbasierte Entwicklungsmethode. Der Kurs führt in wesentliche Vorgehensmodelle für die Softwareentwicklung ein. Es werden Methoden für die Softwareentwicklung und Qualitätssicherung vermittelt, die innerhalb der Vorgehensmodelle zum Einsatz kommen. Außerdem werden Modellierungssprachen vorgestellt, mit denen die statischen und dynamischen Aspekte von Softwaresystemen beschrieben werden können. Hierzu wird die objektorientierte Modellierungssprache UML (Unified Modeling Language) eingeführt, die dedizierte Diagrammsprachen vereint. Methodische Hinweise zum Einsatz der Modellierungssprachen im Softwareentwicklungsprozess, zur Erstellung der Ergebnisartefakte und zur Prüfung ihrer Qualität werden gegeben. Modellierungswerkzeuge werden exemplarisch eingesetzt. Schließlich werden spezifische Programmierkonzepte und werkzeuge für die Softwareentwicklung im Team betrachtet. Die Veranstaltung umfasst folgende Inhalte: • Einführung in die Konzepte und Begrifflichkeiten der Softwaretechnik (Software Engineering) • Softwarelebenszyklus und Vorgehensmodelle (klassische, agile, hybride) • Modellierungssprachen und modellbasierte Softwareentwicklung • UML (Unified Modeling Language): Klassendiagramme, Use-Case-Diagramme, Aktivitätendiagramme, Sequenzdiagramme, Zustandsdiagramme, Komponentendiagramme • durchgängige Softwareentwicklungsmethode von der Anforderungsspezifikation über Modellierung (Softwarearchitektur, Softwareentwurf) bis zur Implementierung und dem Test der Software • Transformation und Konsistenz von Modellen, Codegenerierung • spezifische Programmierkonzepte (Objektorientierung, Typisierung, Schnittstellen) für die arbeitsteilige Softwareentwicklung im Team • Softwaretesten und modellbasiertes Testen • Konzepte und Werkzeuge für Quellcode-Verwaltung, Continuous Integration/Delivery

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können nach Abschluss des Moduls

- für eine gegebene Problemstellung systematisch und schrittweise eine Softwarelösung unter Einhaltung eines Prozessmodells und Einsatz einer (modellbasierten) Entwicklungsmethode entwickeln
- die Unterschiede von Vorgehensmodellen erklären, die Beziehungen und Übergänge zwischen verschieden Entwicklungsphasen eines Vorgehensmodells verstehen und für ein Entwicklungsprojekt ein geeignetes Vorgehen wählen
- wichtige Modellierungssprachen, insbesondere die verschiedenen Diagrammsprachen der UML, für die einzelnen Entwicklungsschritte bei der Modellierung der unterschiedlichen Aspekte einer Softwarelösung auswählen und verwenden, um korrekte und passende Softwaremodelle zu erstellen
- die Qualität von Softwaremodellen und (Zwischen-) Ergebnissen bewerten
- zur Überprüfung der Qualität der entwickelten Softwarelösung Techniken des Softwaretestens und des modellbasierten Testens einsetzen
- Konzepte und Softwareentwicklungswerkzeuge für die arbeitsteilige Softwareentwicklung im Team auswählen und einsetzen
- eine größere, zusammenhängende Softwaremodellierungsaufgabe über einen längeren Zeitraum selbstorganisiert im Team bearbeiten und hierbei die Bedeutung der verschiedenen Phasen, Methoden und Softwarewerkzeuge verstehen

6 **Prüfungsleistung:**

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

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	90-120 min	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 der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).

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

Bachelorstudiengang Informatik v5 (IBA v5), Masterstudiengang Nachhaltiger Maschinenbau V1

12 **Modulbeauftragte/r:**

Dr. Stefan Sauer, Prof. Dr. Eric Bodden

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 (Zentralübung, Präsenz- und/oder Heimübungen in Kleingruppen) vertieft und in dem begleitenden Praktikumsanteil im Team an einem durchgängigen Beispiel von den Studierenden selbst erprobt werden. Hierbei kommen gängige Software-Engineering-Werkzeuge zum Einsatz. **Lernmaterialien, Literaturangaben** • Präsentationsfolien, ggf. Tafelanschrieb, i.d.R. Vorlesungsaufzeichnung • Übungsaufgaben und Lösungshinweise • Praktikumsaufgabe mit Beispielartefakten • Konkrete Literatur zu den einzelnen Themengebieten wird in der Veranstaltung bekanntgegeben.

Programmiersprachen			
Programming Languages			
Modulnummer: M.079.1124	**Workload (h):** 120	**Leistungspunkte:** 4	**Turnus:** module.frequency.de
	Studiensemester: 3	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05301 Programmiersprachen	V2 Z1 Ü1	60	60	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Programmiersprachen:
Empfohlene Vorkenntnisse
Kenntnisse in der Programmierung werden in dem Umfang vorausgesetzt, wie sie im Modul *Programmierung 1* gelehrt werden. Das Modul *Programmiersprachen* ist ausdrücklich auch für Studierende im ersten Semester geeignet, sofern diese bereits sehr gute Programmier-Kenntnisse und Erfahrung in Programmierung haben.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Programmiersprachen:* Das Modul *Programmiersprachen* erweitert den Inhalt des Moduls *Programmierung 1*, in der eine imperativen Programmiersprache gelehrt und zur Programmierung von Algorithmen genutzt wird. Das Modul *Programmiersprachen* baut hierauf auf und vermittelt Erfahrungen in anderen wichtigen Programmierparadigmen, nämlich in der Logischen Programmierung sowie in der Funktionalen Programmierung. Im Zentrum dieses Moduls liegt das praktische Erlernen dieser beiden neuen Programmier-Paradigmen und deren praktische Umsetzung an konkreten Aufgaben in jeweils einer Logikprogrammiersprache bzw. einer funktionalen Programmiersprache. Inhaltsübersicht: Logische Programmierung: Grundlegende Konzepte der Logischen Programmierung, insbesondere • Fakten und Regeln • Choice-Points und Backtracking • Linksrekursion und Rechtsrekursion • Vertauschbarkeit von Eingabe und Ausgabe • Typische Anwendungen (z.B. Zahlenrätsel, Logik-Rätsel, Suche, . . .) • Grammatiken und Term-Ersetzungssysteme Funktionale Programmierung: Grundlegende Konzepte der Funktionalen Programmierung, insbesondere • Seiteneffekte und Programmieren ohne Seiteneffekte • rekursive Programmierung mit Listen • Funktionen höherer Ordnung • Anwendungen und Beispiele, die die Stärke funktionaler Programmierung zeigen
5	**Lernergebnisse und Kompetenzen:** Nach Abschluss dieses Moduls können Studierende • die grundlegenden syntaktischen und semantischen Konzepte und typischen Eigenschaften von Logischer und Funktionaler Programmierung erklären. • die Stärken, Schwächen und Herausforderungen der verschiedenen Programmierparadigmen erläutern und ein geeignetes Programmierparadigma für eine gegebenes Problem auswählen. • Programme und die darin verwendeten Sprachkonstrukte in präziser Weise erläutern. • Programme zur Lösung einfacher Probleme oder zur Implementierung grundlegender Datenstrukturen in einer funktionalen Programmiersprache und in einer Logikprogrammiersprache entwerfen, implementieren und testen. • neue Programmier- und Anwendungssprachen, die den gelehrten Programmierparadigmen angehören, selbstständig erlernen. • mit anderen Studierenden in Kleingruppen zusammenarbeiten, um gemeinsam eine geeignete Lösung für ein gegebenes Programmierproblem zu entwickeln und zu implementieren.

6 **Prüfungsleistung:**

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

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

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben oder Kurzklausur		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 Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5), Bachelorstudiengang Nachhaltiger Maschinenbau V1, Bachelorstudiengang Nachhaltiger Maschinenbau V1b

12 **Modulbeauftragte/r:**

Dr. Rita Hartel, Prof. Dr. Stefan Böttcher

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Programmiersprachen:
Methodische Umsetzung
Die Inhalte werden in einer Präsentation im Rahmen einer Vorlesung eingeführt und anschließend in Präsenzübungen in Kleingruppen sowie in Übungsaufgaben vertieft und durch praktische Übungen ergänzt.
Lernmaterialien, Literaturangaben

- Michael Sperber, Herbert Klaeren: Schreibe Dein Programm! Einführung in die Programmierung. https://www.deinprogramm.de
- Max Bramer: Logic Programming with Prolog, 2nd edition 2013, Springer.
- Iwan Bratko: Prolog Programming for Artificial Intelligence, neueste Ausgabe.
- Peter Flach: Simply Logical. https://book.simply-logical.space/src/simply-logical.html

Datenstrukturen und Algorithmen			
Data Structures and Algorithms			
Modulnummer: M.079.1222	**Workload (h):** 240	**Leistungspunkte:** 8	**Turnus:** module.frequency.de
	Studiensemester: 3	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.079.05302 Datenstrukturen und Algorithmen	V4 Ü2 Z1	105	135	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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

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

5 **Lernergebnisse und Kompetenzen:**

Nach Abschluss des Moduls können die Studierenden

- effiziente Datenstrukturen und Algorithmen für ausgewählte grundlegende Probleme erklären und anwenden,
- Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen einsetzen,
- selbstständig und kreativ Algorithmen und Datenstrukturen entwickeln,
- mathematische Methoden zum Korrektheitsbeweis und zur Effizienzanalyse einsetzen,
- die Wechselwirkung zwischen Algorithmus und Datenstruktur an wesentlichen Beispielen erläutern,
- die Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten einschätzen,
- sich neue Algorithmen, Datenstrukturen und algorithmische Ideen und Analysen aneignen.

6 **Prüfungsleistung:**

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

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

<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>
</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 Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5), Masterstudiengang Nachhaltiger Maschinenbau V1</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Christian Scheideler, Prof. Dr. Sevag Gharibian, Prof. Dr. Stefan Böttcher</td></tr>
<tr><td>13</td><td>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</td></tr>
</table>

Rechnerarchitektur			
Computer Architecture			
Modulnummer: M.079.1322	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 3	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05303 Rechnerarchitektur	V2 Ü2	60	120	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Rechnerarchitektur:
Empfohlene Vorkenntnisse
Kenntnisse aus dem Modul *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.
Die Veranstaltung umfasst folgende Inhalte:

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

5	**Lernergebnisse und Kompetenzen:** Die Studierenden können • den Aufbau eines modernen Rechners sowie das Zusammenspiel von Hardware und Software beschreiben, • die zugrunde liegenden allgemeinen Entwurfsprinzipien und -strategien erklären und anwenden, • Rechnersysteme im Hinblick auf Leistung und Kosten analysieren und bewerten, sowie • selbständig einfache Assemblerprogramme schreiben.
6	**Prüfungsleistung:** ☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	90-120 min	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 der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v5 (IBA v5), Masterstudiengang Nachhaltiger Maschinenbau V1
12	**Modulbeauftragte/r:** Prof. Dr. Christian Plessl, Prof. Dr. Marco Platzner

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Rechnerarchitektur:* **Methodische Umsetzung** Die Lehrveranstaltung besteht aus einer Vorlesung und Rechen- sowie praktischen Übungen. Die Vorlesung erfolgt mit Beamer und Tafelanschrieb. In den Rechenübungen werden Aufgaben ausgegeben und danach werden im Rahmen von Präsenzübungen in Kleingruppen die Lösungen durch die Übungsteilnehmer vorgestellt und diskutiert. In den praktischen Übungen werden Aufgaben ausgegeben und von den Übungsteilnehmern in Kleingruppen als Assemblerprogramme umgesetzt. **Lernmaterialien, Literaturangaben** • Vorlesungsfolien und Aufgabenblätter für Rechenübungen • Aufgabenblätter für praktische Übungen und technische Dokumentation zur Assemblerprogrammierung • D. A. Patterson, J. L. Hennessy: Computer Organization & Design – The Hardware / Software Interface (RISC-V Edition); Morgan Kaufmann, 2018 • Aktuelle Hinweise auf ergänzende Literatur und Lehrmaterialien werden in der Vorlesung bekannt gegeben.

Stochastik für Informatik

Stochastic for Computer Science

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.105.96300	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	3	1	de

1 **Modulstruktur**

	Lehrveranstaltung	**Lehrform**	**Kontaktzeit (h)**	**Selbststudium (h)**	**Status (P/WP)**	**Gruppengröße (TN)**
a)	L.105.97300 Stochastik für Informatik	V3 Ü2	75	105	P	200/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Stochastik für Informatik:
Empfohlene Vorkenntnisse
Mindestens eines der Module *Analysis für Informatik* oder *Lineare Algebra für Informatik* sollte erfolgreich abgeschlossen sein.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Stochastik für Informatik:
Deskriptive Statistik und Datenanalyse

- statistische Erhebungen (Befragung, Beobachtung oder Experiment)
- Darstellungen für uni- und bivariate Daten (z.B. Kreuztabelle)

Zufallsmodellierung

- mehrstufige Zufallsversuche mit endlichen Ergebnismengen
- Rechnen und Argumentieren mit Wahrscheinlichkeiten, bedingten Wahrscheinlichkeiten, Erwartungswerten und stochastischer Unabhängigkeit
- Bernoullisches Gesetz der großen Zahlen und zentraler Grenzwertsatz
- diskrete und kontinuierliche Verteilungen uns deren Einsatz in der Modellierung

Stochastische Anwendungen

- Anwendung von Stochastik in verschiedenen Wissenschaften (Ökonomie, Physik, . . .)
- Parameterschätzungen aus Daten in Zufallssituationen
- Hypothesentests
- Bayes-Statistik und der Vergleich zu klassischen Testverfahren

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden
• kennen die Bedeutung der Stochastik,
• können sicher mit den grundlegenden Begriffen, Methoden und Theoremen der Stochastik umgehen,
• besitzen Kompetenzen in der stochastischen Modellierung,
• verstehen Beweise,
• können Übungsaufgaben zur Stochastik lösen,
• können Verbindungen innerhalb der Stochastik beziehungsweise zwischen der Stochastik und anderen Bereichen der Mathematik erkennen,
• können einfache statistische Analysen durchführen.</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>120-180 min</td><td>100%</td></tr></table></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 Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Martin Kolb</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Stochastik für Informatik:* **Methodische Umsetzung** In der Vorlesung werden • die grundlegenden Begriffe eingeführt, • mathematische Konzepte formuliert, analysiert und in Teilen bewiesen • die theoretischen Inhalte anhand einfacher Beispiele veranschaulicht In den Übungen werden • die erlernten Konzepte umfassend auf Fragestellungen verschiedener Komplexität angewendet • weiter führende Beispiele betrachtet, um das Erlernte zu erweitern bzw. um weitere Perspektiven zu ergänzen Die Studienleistung • dient der besseren Vorbereitung auf die Abschlussprüfung • motiviert Studierende, kontinuierlich mitzuarbeiten • ermöglicht regelmäßige Rückmeldungen über den eigenen Lernfortschritt

2.4 4. Semester

Softwaretechnikpraktikum

Software Engineering Project

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.1125	240	8	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	4	1	de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05400 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 1* 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

<table>
<tr><td>5</td><td>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.</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>Software mit Dokumentation, Projektpräsentation, schriftliche Hausarbeit</td><td>i.d.R. bis zu 2500 Zeilen Code, bis zu 4 Seiten, 30 min, bis zu 10 Seiten</td><td>45%, 25%, 30%</td></tr>
</table>
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%.</td></tr>
<tr><td>7</td><td>Studienleistung, qualifizierte Teilnahme:
keine</td></tr>
<tr><td>8</td><td>Voraussetzungen für die Teilnahme an Prüfungen:
Die regelmäßige Teilnahme an den Übungen der jeweiligen Gruppe ist Voraussetzung für die Teilnahme an den Modulprüfungen. Zwei Fehltermine sind erlaubt.</td></tr>
<tr><td>9</td><td>Voraussetzungen für die Vergabe von Credits:
Die Vergabe von Credits erfolgt, wenn die Modulteilprüfungen bestanden sind.</td></tr>
<tr><td>10</td><td>Gewichtung für Gesamtnote:
Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Eric Bodden</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Softwaretechnikpraktikum:* **Methodische Umsetzung** Durchführung eines Projekts mit regelmäßiger Abgabe von Arbeitsergebnissen (Modulteilprüfungen), protokollierten Gruppensitzungen und einer Projektpräsentation. **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.

Berechenbarkeit und Komplexität			
Computability and Complexity			
Modulnummer: M.079.1223	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 4	**Dauer (in Sem.):** 1	**Sprache:** de

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05402 Berechenbarkeit und Komplexität	V3 Ü2	75	105	P	200/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Berechenbarkeit und Komplexität:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Modellierung* und *Datenstrukturen und Algorithmen* sind hilfreich.

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.
Die Veranstaltung umfasst folgende Inhalte:

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

5	**Lernergebnisse und Kompetenzen:** Die Studierenden können • grundlegende Literatur im Bereich der Berechenbarkeit und Komplexitätstheorie verstehen, • selbständig Probleme in diesem Bereich analysieren und klassifizieren, • Hypothesen zur Komplexität von Problemen entwickeln und diese anschließend verifizieren oder falsifizieren und • darauf aufbauend neue Hypothesen formulieren.
6	**Prüfungsleistung:** ☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

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

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben		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 Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Prof. Dr. Christian Scheideler, Prof. Dr. Johannes Blömer, Prof. Dr. Sevag Gharibian

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 eine Zentralübung 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

Betriebssysteme			
Operating Systems			
Modulnummer: M.079.1324	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 4	**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.079.05404 Betriebssysteme	V2 Ü2	60	120	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Betriebssysteme:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Modellierung* und *Rechnerarchitektur* sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Betriebssysteme:
Die Vorlesung Betriebssysteme behandelt grundlegende konzeptionelle und technologische Grundlagen von Betriebssystemen. Die Veranstaltung umfasst folgende Inhalte:

- Einführung in Betriebssysteme
- Zusammenfassung Rechnerarchitektur
- Prozesse und Threads
- Synchronisation und Interprozess-Kommunikation
- Ablaufplanung
- Speicherverwaltung
- Virtualisierung
- Aspekte der Sicherheit

5	**Lernergebnisse und Kompetenzen:** Die Studierenden können • die grundlegenden Konzepte von und Anforderungen an Betriebssysteme beschreiben, • Synchronisations- und Kommunikationsmechanismen von Prozessen und Threads aufzählen, deren Eigenschaften vergleichen und an einfachen Programmbeispielen anwenden, • Methoden der Ablaufplanung erklären und an einfachen Programmbeispielen anwenden, • das Zusammenspiel von Speicherverwaltung Prozess- bzw. Threadverwaltung erklären und • die Aspekte von Virtualisierung und Sicherheit auf Ebene der Betriebssysteme diskutieren und einordnen.
6	**Prüfungsleistung:** ☒Modulabschlussprüfung (MAP) □Modulprüfung (MP) □Modulteilprüfungen (MTP)

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a)	Klausur	90-120 min	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 der Anzahl seiner Leistungspunkte gewichtet (Faktor 1).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Prof. Dr. Christian Plessl, Prof. Dr. Marco Platzner
13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Betriebssysteme:* **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 • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

IT-Sicherheit			
IT Security			
Modulnummer: M.079.1325	**Workload (h):** 150	**Leistungspunkte:** 5	**Turnus:** module.frequency.de
	Studiensemester: 4	**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.079.05403 IT-Sicherheit	V2 Ü2	60	90	P	150/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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.
Die Veranstaltung umfasst folgende Inhalte:

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

5 Lernergebnisse und Kompetenzen:

Die Studierenden können

- die wesentlichen Konzepte, Methoden und Mechanismen zum Schutz von Daten und Systemen vor Manipulation und Missbrauch verstehen
- Konzepte zur Erhöhung der Systemsicherheit korrekt einsetzen
- einfache Sicherheitsprotokolle entwickeln und bewerten
- die Ursachen von Sicherheits-Problemen heutiger Systeme verstehen
- praktische Sicherheitsevaluierung von grundlegenden Applikationen und kryptographischen Protokollen durchführen

6 Prüfungsleistung:

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

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

7 Studienleistung, qualifizierte Teilnahme:

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben		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 Modulabschlussprüfung bestanden ist.

10 Gewichtung für Gesamtnote:

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

11 Verwendung des Moduls in anderen Studiengängen:

Bachelorstudiengang Informatik v5 (IBA v5), Bachelorstudiengang Nachhaltiger Maschinenbau V1, Bachelorstudiengang Nachhaltiger Maschinenbau V1b

12 Modulbeauftragte/r:

Prof. Dr.-Ing. Juraj Somorovsky

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung IT-Sicherheit:* **Methodische Umsetzung:** Die Inhalte werden durch eine Präsentation in Form einer Vorlesung mit Beamer vermittelt. Dazu werden sie in Präsenzübungen in Kleingruppen, sowie durch individuelle praktische Aufgaben vertieft. **Lernmaterialien, Literaturangaben:** • Vorlesungsfolien und Übungsblätter • Wissenschaftliche Literatur • Sicherheit und Kryptographie im Internet, Jörg Schwenk • Computer Security, William Stallings und Lawrie Brown • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Informatik und Gesellschaft			
Computer Science and Society			
Modulnummer: M.079.1522	**Workload (h):** 150	**Leistungspunkte:** 5	**Turnus:** module.frequency.de
	Studiensemester: 4	**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.079.05401 Informatik und Gesellschaft	V3 Ü2	75	75	P	200/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**
keine

3 **Teilnahmevoraussetzungen:**
keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Informatik und Gesellschaft:
Die Veranstaltung gibt einen Überblick über die gesellschaftlichen Aspekte der Informationstechnik und versetzt die Studierenden in die Lage, die gesellschaftliche und ethische Bedeutung des Fachs zu beurteilen und Konsequenzen für verantwortungsbewusstes Handeln zu ziehen. Die Studierenden verstehen die Zusammenhänge zwischen Informationstechnik und Gesellschaft, sind in der Lage, die Auswirkungen informationstechnischer Produkte und Dienstleistungen zu analysieren und potentielle Konfliktfelder zu identifizieren, und kennen ethische Verhaltensregeln, wie sie zum Beispiel von den Berufsorganisationen GI, IEEE und ACM herausgegeben werden.
Die Veranstaltung umfasst folgende Inhalte:

- Geschichte der Informatik: Geschichte des Schreibens, Rechnens und Kommunizierens, Wechselwirkungen zwischen der Entwicklung von Gesellschaft und Technologien, Charakteristika und Potenziale des Digitalen, Informationstechnologie zwischen Herrschaftsinstrument oder Empowerment
- Einführung in Ethik: Ethische Theorien, Theorie der Gerechtigkeit, Ethische Leitlinien
- Vernetzte Kommunikation: Wissensgesellschaft, Informationsfreiheit, Meinungsfreiheit und Zensur, Chatkontrolle, Netzneutralität, Störerhaftung
- Geistiges Eigentum: Urheberrecht, Digital Rights Management, Markenrecht, Patentrecht, Software als Produkt, Free and Open Software, Creative Commons
- Datenschutz: Entstehung des Datenschutz, Datenschutz vs. Privatsphäre, Grundkonzepte des Datenschutzes, Grundsätze der DSGVO, Rechte der Betroffenen, Technisch-organisatorische Maßnahmen, Auftragsverarbeitung, Datenschutz-Management
- Datenschutzgefährdende Technologien: Datenspuren im Netz und in der realen Welt, Datenspeicherung im Ausland, Gesichtserkennung, Social Profiling, Predictive Policing, RFID-Technologie, Surveillance Capitalism, Big Data, De-Pseudonymisierung
- Informatik und das Militär: Militärgeschichte der Informatik, Kernprobleme im militärischen Bereich, Besonderheiten von Software, Fehler und Zuverlässigkeit von Computern, Autonome Waffensysteme, Cyberwar

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- ethische Herausforderungen in der Arbeit von Informatikerinnen und Informatikern erkennen,
- Grundkonzepte der behandelten Themenfelder (Geschichte, Ethik, Datenschutz, geistiges Eigentum etc.) erklären,
- Systemanforderungen auf Grundlage der ethischen und rechtlichen Rahmenbedingungen formulieren,
- Wechselwirkungen zwischen gesellschaftlichen Anforderungen, rechtlichen Rahmenbedingungen und Informatiksystemen analysieren,
- informatische Aspekte in gesellschaftlich relevanten Ereignissen bewerten.

6 **Prüfungsleistung:**

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

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

<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></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 Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5), Bachelorstudiengang Nachhaltiger Maschinenbau V1, Bachelorstudiengang Nachhaltiger Maschinenbau V1b</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Dr. Harald Selke</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Informatik und Gesellschaft:
Methodische Umsetzung
Die Inhalte werden durch den Dozenten im Rahmen der Vorlesung präsentiert, wobei interaktive Elemente sowie punktuell das Flipped-Classroom-Konzept zur Anwendung kommen. In wöchentlich zu bearbeitenden Übungsaufgaben werden Fallbeispiele und Szenarien erarbeitet. In den Übungen diskutieren die Studierenden in Kleingruppen und referieren in den Übungen sowie in den Übungsaufgaben erarbeitete Positionen und Lösungsvorschläge.
Lernmaterialien, Literaturangaben
• Michael J. Quinn: Ethics for the Information Age. 9th edition, Pearson, 2025.
• Sara Baase, Timothy M. Henry: A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology. 5th edition, Pearson, 2018.
• Felix Winkelnkemper: Interface Evolution – Die Geschichte des Computers als Geschichte seiner Nutzungsschnittstelle. Eigenverlag, 2021.
• Weitere Literatur wird in der Veranstaltung bekanntgegeben.</td></tr>
</table>

2.5 5. Semester

Einführung in Data Science			
Introduction to Data Science			
Modulnummer: M.079.1422	**Workload (h):** 150	**Leistungspunkte:** 5	**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)	2024.5002 Einführung in Data Science	V2 Ü2	60	90	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Einführung in Data Science:
Eine zunehmende Anzahl von Geräten des täglichen Gebrauchs wird von datengetriebenen Verfahren gesteuert. In diesem Modul werden die Grundlagen wichtiger datengetriebener Verfahren vermittelt. Ausgangspunkt für dieses Modul werden grundlegende Definitionen und Algorithmen bilden. Darauf aufbauend werden Implementierungen und Anwendungen dieser Verfahren diskutiert. Die praktischen Übungen mit echten Daten werden die teilnehmenden Studierenden in die Lage versetzen, praktische Fragestellungen mit den besprochenen Verfahren zu adressieren.
Die Lehrveranstaltung umfasst folgende Inhalte:

- Das Messproblem
- Daten repräsentieren
- Daten akquirieren
- Daten reinigen
- Daten visualisieren
- Unüberwachte Verfahren
- Überwachte Verfahren
- Anwendungen

5 **Lernergebnisse und Kompetenzen:**

Nach Abschluss dieses Moduls können die Studierenden

- Messfehler in Daten modellieren,
- Daten anwendungsgerecht modellieren,
- Messfehler bei der Anwendung von datengetriebenen Verfahren berücksichtigen,
- Verfahren evaluieren,
- Verfahren zur Reinigung von Daten erklären, implementieren und anwenden,
- einfache überwachte und unüberwachte Verfahren erklären, implementieren und anwenden,
- Implementierungen einfacher Verfahren optimieren.

6 **Prüfungsleistung:**

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

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

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben		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 Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Axel-Cyrille Ngonga Ngomo

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Einführung in Data Science:* **Methodische Umsetzung** In den wöchentlichen Vorlesungen werden neue Inhalte behandelt. Die Übungen sind sowohl theoretischer als auch praktischer Natur. Die Lernenden sollen zeigen, dass sie die Konzepte verstanden haben und sie auf praktische Probleme anwenden können. **Lernmaterialien, Literaturangaben** • Tom Mitchell: “Machine Learning”. McGraw-Hill, 1997 • Christopher Bishop: “Pattern Recognition and Machine Learning”. Springer, 2006 • Christopher Manning, Hinrich Schütze. “Foundations of Statistical Natural Language Processing”, 2022

Rechnernetze			
Computer Networks			
Modulnummer: M.079.1323	**Workload (h):** 150	**Leistungspunkte:** 5	**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.079.05506 Rechnernetze	V2 Ü2	60	90	P	300/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Rechnernetze:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Programmierung 1*, *Programmierung 2*, *Modellierung*, *Digitaltechnik* und *Datenstrukturen und Algorithmen* sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Rechnernetze:
Das Internet ist schon seit Jahrzehnten der Motor unserer digitalen Welt. Es ist auch eine der wichtigsten Erfindungen in der Geschichte der Menschheit. Während jeder das Internet für verschiedene Sachen wie Nachrichten, soziale Netzwerke und Videokonferenzen nutzt, verstehen nur wenige, wie es wirklich funktioniert. In diesem Kurs geht es um die Details des Internets und die wichtigen Schichten in seiner Architektur. Wir schauen uns vor allem diese Konzepte an:

- Physikalische Schicht: Übertragungsmedium, Signal, Codierung und Modulation, grundlegende Grenzen
- Datenverbindungsschicht: Framing, Fehlererkennung und -korrektur, drahtlose Netzwerke, Ethernet
- Netzwerkschicht: Adressierung, IP-Weiterleitung, Routing-Algorithmen, OSPF, BGP
- Transportschicht: Zuverlässigkeit, UDP, TCP, Überlastungskontrolle
- Anwendungsschicht: DNS, Web, Video-Streaming, Linux-Netzwerkstack

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden sind in der Lage,
• die Grundprinzipien von Computernetzwerken zu erklären (Wissen und Verständnis)
• die Schichtenarchitektur zu beschreiben und die wichtigsten Funktionen in diesen Schichten zu erklären (Wissen und Verständnis)
• Grundlegende Physik, Informationstheorie und Eigenschaften von Netzwerkprotokollen anzuwenden, um Eigenschaften von Netzwerken abzuleiten (Anwendung von Wissen und Verständnis)
• die Vorteile und Grenzen der heute gängigen Netzwerkprotokolle zu erklären (Urteilsvermögen)
• ihre Kenntnisse in der Netzwerkprogrammierung durch die Implementierung von Netzwerkanwendungen unter Beweis zu stellen (Anwendung von Wissen und Verständnis)</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></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 Leistungspunkte gewichtet (Faktor 1).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Lin Wang</td></tr>
</table>

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Rechnernetze:* **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: Computer Networks (Tanenbaum, Wetherall, Feamster), Computer Networking: A Top-Down Approach (Kurose, Ross) • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

3 Vertiefungsstudium

Im Vertiefungsstudium haben die Studierenden die Gelegenheit, sich in Fächern ihrer Wahl gezielt zu vertiefen. Gleichzeitig wird weiterhin die Breite des Studiums garantiert, indem alle Studierenden weiterführende Veranstaltungen aus mindestens zwei der vier Gebiete 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

Koordination: Prof. Dr. Christian Scheideler

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 Gebiets 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			
3D Modeling			
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:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

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

3D Rendering

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2231	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.079.05641 3D Rendering	V3 Ü2	75	105	P	60/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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

5 **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

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%

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Fortschrittsberichte		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 Modulabschlussprüfung bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Marcel Campen

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

Introduction to Cryptography

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.079.05638 Einführung in Kryptographie	V3 Ü2	75	105	P	100/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Kryptographie:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Datenstrukturen und Algorithmen* sowie *Berechenbarkeit und Komplexität* sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Einführung in Kryptographie:
Kryptographie ist eine wichtige Basistechnik der IT-Sicherheit. So beruhen Internet-Protokolle wie TLS auf kryptgraphischen Primitiven wie Schlüsselaustausch, Verschlüsselung und Signaturen. 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.
Die Veranstaltung umfasst folgende Inhalte:

- 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:** Nach Abschluss des Moduls können die Studierenden • Sicherheitsanforderungen mittels kryptographischer Aufgaben formulieren. • die wichtigsten kryptographischen Basistechniken und ihre Einsatzmöglichkeiten. • einschätzen, ob umgesetzte kryptographische Lösungen gegebenen Anforderungen genügen • können für gegebene Sicherheitsanforderungen die geeigneten kryptographischen Verfahren auswählen. • können einschätzen, welche Anpassungen an kryptographische Verfahren unproblematisch sind und welche sicherheitskritisch sind.
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%

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		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 Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Prof. Dr. Johannes Blömer

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Einführung in Kryptographie:* **Methodische Umsetzung** Grundlegende Konzepte werden in einer Vorlesung präsentiert. Zusätzlich werden theoretische Konzepte in Tutorien in Kleingruppen vertieft. In schriftlichen Übungen und in Lesegruppen wird der praktischer Einsatz dieser Konzepte eingeübt. **Lernmaterialien, Literaturangaben** • Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography, Chapman and Hall, • Johannes Buchmann: Einführung in Kryptographie, Springer Verlag, • Dan Boneh, Victor Shoup: A Graduate Course in Applied Cryptography: https://crypto.stanford.edu/~dabo/cryptobook/, • Vorlesungsfolien, Übungsaufgaben

Grundlagen der Algorithmischen Geometrie			
Fundamentals of Computational Geometry			
Modulnummer: M.079.2215	**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.079.05603 Grundlagen der Algorithmischen Geometrie	V3 Ü2	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Grundlagen der Algorithmischen Geometrie:
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 Analyse wird angenommen.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Grundlagen der Algorithmischen Geometrie:
Es werden grundlegende Algorithmen und Datenstrukturen aus dem Bereich der algorithmischen Geometrie behandelt. Die Grundelemente der Eingabe sind geometrische Daten. Die Probleme werden geometrisch formuliert. Zur Lösung der Probleme werden Algorithmen gesucht und entwickelt, die in der Regel mit Datenstrukturen arbeiten. Die Datenstrukturen können klassische Datenstrukturen oder spezielle geometrische oder räumliche Datenstrukturen sein.
Der Sweep-Ansatz wird als grundlegendes algorithmisches Programmier-Paradigma eingeführt. Dieser stellt als Sweep-Line- oder Sweep-Plane-Ansatz eine Schlüsseltechnik der algorithmischen Geometrie dar. Er wird an verschiedenen Problemen demonstriert.
Die Algorithmen werden theoretisch analysiert. Dazu wird Laufzeit und Speicherplatz bestimmt und die Korrektheit der Algorithmen bewiesen.
Die Veranstaltung umfasst folgende Inhalte:

- Geometrische Daten: Punkte, Linien, Kreise, Polygone, Körper.
- Algorithmen für geometrische Probleme: Bereichssuche, Schnittprobleme, Nachbarschaftssuche, Bewegungsplanung, Sichtbarkeit, konvexe Hüllen, Konturberechnungen.
- Datenstrukturen zur Lösung geometrischer Probleme: Voronoi-Diagramme, Delaunay-Triangulierung, Bereichsbäume, k-d-Bäume, BSP-Bäume, Prioritätssuchbäume.
- Sweep-Line-Algorithmen: Schnitt von Liniensegmenten, dichtestes Punktepaar, Sichtbarkeit

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- grundlegende Algorithmen und Datenstrukturen für Problemstellungen der algorithmischen Geometrie benennen, erklären und anwenden
- grundlegende algorithmische Problemstellungen in Anwendungsproblemen der algorithmischen Geometrie erkennen und geeignete Algorithmen und Datenstrukturen dafür auswählen
- Laufzeit- und Speicherplatzabschätzung von geometrischen Datenstrukturen und Algorithmen analysieren, vergleichen und untersuchen
- beurteilen, welche Auswirkungen die Wahl von Datenstrukturen auf die Effizienz von Algorithmen der algorithmischen Geometrie hat
- eigene effiziente Algorithmen auf Basis des Sweep-Line Paradigmas für weitere geometrische Probleme entwickeln
- eigene effiziente Algorithmen auf Basis des Voronoi-Diagramms für weitere geometrische Probleme entwickeln, die sich mit dieser Struktur geeignet lösen lassen
- Problemstellungen der algorithmischen Geometrie und entsprechende Lösungsvorschläge mit Fachexperten diskutieren

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Dr. Matthias Fischer

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Grundlagen der Algorithmischen Geometrie:
Methodische Umsetzung
Die Inhalte werden durch eine Präsentation im Rahmen einer Vorlesung vermittelt. Die Vorlesung erfolgt in der Regel mit Beamer und Tafelanschrieb. In Übungen und Übungsaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen vorgestellt und von den Studierenden in Präsenzübungen in Kleingruppen sowie in Übungsaufgaben vertieft und selbst entwickelt und durch praktische Übungen ergänzt. Musterlösungen von Übungsblättern werden in Zentralübungen vorgestellt. Die erwarteten Aktivitäten der Studierenden sind die Mitarbeit bei Präsenzübungen und das selbstständige Bearbeiten der Übungsaufgaben.
Lernmaterialien

- Vorlesungsfolien, Übungsblätter, ggfs. Musterlösungen, Vorlesungsaufzeichnung aus früheren Jahren, Tafelanschrieb

Literaturangaben

- Algorithmische Geometrie, Rolf Klein, Springer-Verlag, 2005.
- Computational Geometry - An Introduction, Franco P. Preparata; Springer, 1993.
- Lectures on Discrete Geomtetry, Jiri Matousek, Springer-Verlag, 2002.
- Computational Geometry: Algorithms and Applications, Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars, Springer-Verlag, 2008.
- Handbook of Discrete and Computational Geometry, Jacob E. Goodman, Joseph O'Rourke, CRC Press, 1997.
- Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Grundlegende Algorithmen

Fundamental Algorithms

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.079.05502 Grundlegende Algorithmen	V3 Ü2	75	105	P	75/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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 empfohlen.

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 sind in der Lage:

- Algorithmen für fortgeschrittene Probleme zu entwerfen
- zu erkennen, wann Probleme mit Hilfe bestehender Algorithmen gelöst werden können, wie z. B. Netzwerkfluss und lineare Programmierung
- die Korrektheit von Algorithmen rigoros zu beweisen
- Strenge Analyse von Algorithmus-Laufzeiten
- Verwendung der Dualität der linearen Programmierung für theoretische Analysen

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)

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. • Übungsblätter, Lösungen werden in Übungsgruppen vorgestellt und diskutiert. • In Übungen und Übungsaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen geübt. **Lernmaterialien, Literaturangaben** • Foliensatz der Vorlesung, Übungsblätter • T.H. Corman, C.E.Leiserson, R.L. Rivest, C. Stein. Introduction to Algorithms. MIT Press, 2002. • J. Kleinberg, E. Tardos. Algorithm Design. Pearson, 2006.

Komplexitätstheorie			
Complexity Theory			
Modulnummer: M.079.2212	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**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.079.05612 Komplexitätstheorie	V3 Ü2	75	105	P	30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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:**

Die Studierenden können

- weiterführende Literatur zur Komplexitätstheorie nachvollziehen und einordnen,
- die Bedeutung und Unterschiede unterschiedlicher Komplexitätsklassen wie PSPACE, NL, PH und BPP verstehen und erläutern,
- Hypothesen zum Verhältnis von Komplexitätsklassen formulieren und Begründen
- Komplexitätsklassen nutzen, um die Komplexität von Problemen einzuordnen und zu vergleichen
- unterschiedliche Komplexitätsmaße wie Zeit, Platz, Kommunikation und Zufall auf die Einordnung der Komplexität von Problemen anwenden

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

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

Quantum Networks

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2230	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5	1	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:**

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, mündliche Prüfung oder Referat	120-180 min, 30-45 min oder 30 min	100%

<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></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 v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Dr. Zahra Raissi</td></tr>
<tr><td>13</td><td>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</td></tr>
</table>

Verteilte Algorithmen und Datenstrukturen

Distributed Algorithms and Data Structures

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2214	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.079.05610 Verteilte Algorithmen und Datenstrukturen	V3 Ü2	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Verteilte Algorithmen und Datenstrukturen:
Empfohlene Vorkenntnisse
Kenntnisse aus dem Modul *Datenstrukturen und Algorithmen* sind hilfreich.

4 **Inhalte:**

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 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

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- grundlegende Literatur im Bereich verteilte Algorithmen und Datenstrukturen verstehen,
- selbständig adäquate Techniken und Verfahren im Bereich der verteilten Algorithmen und Datenstrukturen anwenden,
- algorithmische Probleme gemäß ihrer Lösbarkeit und Komplexität einschätzen und
- grundlegende verteilte Datenstrukturen implementieren.

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Christian Scheideler

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Verteilte Algorithmen und Datenstrukturen:* **Methodische Umsetzung** Die Vorlesung nutzt Tafelanschrieb und Folien sowie kleine Aufgaben für die Studierenden während der Vorlesung. Sie wird durch Tutorien begleitet. Studierende haben in den Tutorien Gelegenheit, Aufgaben in der Gruppe zu bearbeiten und Lösungen zu den Übungsblättern mit den Tutoren zu diskutieren. **Lernmaterialien, Literaturangaben** • Foliensatz der Vorlesung; Übungsblätter • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

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

Koordination: Prof. Dr. Marco Platzner

Jede Informatikanwendung benötigt ein technisches System, auf dem 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 von Informatiker*innen. Im Gebiet Computersysteme 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 werden abgeleitet. Dabei fokussieren die einzelnen Module, die innerhalb dieses Gebiets gewählt werden können, auf einzelne Teilbereiche von Computersystemen: Vernetzung von Rechnern, verteilte Systeme, Hochleistungsrechner und eingebettete Systeme.

Einführung in Hochleistungsrechnen

Introduction to High-Performance Computing

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	P	25

2	**Wahlmöglichkeiten innerhalb des Moduls:** keine
3	**Teilnahmevoraussetzungen:** keine *Teilnahmevoraussetzungen der Lehrveranstaltung Einführung in Hochleistungsrechnen:* **Empfohlene Vorkenntnisse** Kenntnisse aus dem Modul *Rechnerarchitektur* sind hilfreich.
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)
5	**Lernergebnisse und Kompetenzen:** Nach Abschluss dieses Moduls können Studierende • die grundlegenden Ansätze zur Modellierung, Leistungsbewertung und Optimierung von Rechnersystemen erklären und auf Fallstudien anwenden. • die Ebenen von Parallelverarbeitung in modernen Rechnersystemen benennen, ihre Eigenschaften charakterisieren und daraus Optimierungsstrategien ableiten. • Programmiertechniken zur Parallelverarbeitung beim Entwurf und der Implementierung von Software praktisch anwenden • Parallele Anwendungen auf Hochleistungsrechnern ausführen und eine Leistungsbewertung durchführen.

<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, mündliche Prüfung oder Referat</td><td>90-120 min, 30-45 min oder 30 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:</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 v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr. Christian Plessl</td></tr>
<tr><td>13</td><td>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</td></tr>
</table>

Einführung in Planungs- und Entscheidungsfindungsalgorithmen

Introduction to Planning and Decision-Making Algorithms

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.048.11113	180	6	module.frequency.de
	Studiensemester:	**Dauer (in Sem.):**	**Sprache:**
	5.-6. Semester	1	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.

5 **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.

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 oder 30-45 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

keine

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

Keine

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

Die Vergabe der Leistungspunkte erfolgt, wenn die Modulabschlussprüfung (MAP) bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

11 **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

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			
Embedded Systems			
Modulnummer: M.079.2311	**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.079.05620 Eingebettete Systeme	V3 Ü2	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Eingebettete Systeme:
Empfohlene Vorkenntnisse
Kenntnisse aus den Modulen *Rechnerarchitektur*, *Betriebssysteme* sowie *Programmierung* sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Eingebettete Systeme:
Eingebettete Systeme sind informationsverarbeitende Systeme, die in einen physikalischen Prozess oder eine physikalische Umgebung eingebunden sind. Beispiele dafür finden sich in der Industrieautomatisierung, der Robotik, der Automobiltechnik, in medizintechnischen Produkten und in der Konsumerelektronik. Die verwandten Begriffe Software-intensive Systeme und Cyber-physikalische Systeme beschreiben solche eingebettete Systeme und betonen den zunehmenden Softwareanteil und die Kombination der beiden Aspekte der digitalen Informationsverarbeitung (cyber) und der physikalischen Prozesse (physical). Die Veranstaltung bietet eine Einführung in eingebettete Systeme mit ihren Hardware- und Softwareanteilen und umfasst folgende Inhalte:

- Einführung in eingebettete Systeme
- Spezifikationsmodelle für eingebettete Systeme: zustandsorientierte und datenflussorientierte Modelle
- Zielarchitekturen: General-Purpose Prozessoren, Digitale Signalprozessoren, Mikrokontroller, ASIPs, FPGAs und ASICs, Systems on Chip
- Reaktive und Echtzeitsysteme: Taskdefinitionen, Programmierparadigmen, Echtzeitscheduling, gemeinsam genutzte Ressourcen
- Performance und Energie: Analyse der Worst-case Execution Time (WCET), Energiemetriken, Techniken zur Energieminimierung

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- die Eigenschaften eingebetteter Systeme und ihrer Varianten benennen,
- Spezifikationsmodelle für eingebettete Systeme mit ihren Eigenschaften erklären,
- Ansätze zur Programmierung von Echtzeitsystemen erklären,
- Methoden zur Analyse von Echtzeiteigenschaften für Uniprozessorsysteme auswählen und anwenden,
- Methoden zur Bestimmung der Worst-case execution time auswählen und anwenden und
- die Bedeutung von Performance- und Energiemetriken einschätzen.

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Marco Platzner

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Eingebettete Systeme:* **Methodische Umsetzung** Die Lehrveranstaltung besteht aus einer Vorlesung und Rechen- sowie praktischen Übungen. Die Vorlesung erfolgt mit Beamer und Tafelanschrieb. In den Rechenübungen werden Aufgaben ausgegeben und deren Lösungen während einer Übungseinheit vorgestellt und diskutiert. Zusätzlich werden Quizze zur Lernerfolgskontrolle angeboten. In den praktischen Übungen werden Aufgaben ausgegeben und als Programmierbeispiele mit einem Echtzeitbetriebssystem auf einem eingebetteten Prozessor in Gruppen von einem bis drei Teilnehmern umgesetzt. **Lernmaterialien, Literaturangaben** • Vorlesungsfolien, Aufgabenblätter für Rechenübungen, Quizze • Aufgabenblätter für Programmierbeispiele, technische Dokumentationen • Peter Marwedel: Embedded System Design, 4th edition, Springer, Open Access, 2021 • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Embedded Machine Learning			
Embedded Machine Learning			
Modulnummer: M.079.2331	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 6	**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.079.05648 Embedded Machine Learning	V2 Ü3	75	105	P	100/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

<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></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 Computer Engineering v3 (CEBA v3), Bachelorstudiengang Computer Engineering v3b (CEBA v3b), Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)</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>

Verteilte Systeme			
Distributed Systems			
Modulnummer: M.079.2313	**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.079.05533 Verteilte Systeme	V2 Ü3	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Lin Wang

13	**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.

Ausgewählte Themen im Gebiet Computersysteme

Selected Topics in Computer Systems (BA)

Modulnummer:	Workload (h):	Leistungspunkte:	Turnus:
M.079.2317	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 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

Koordination: Prof. Dr. Axel-Cyrille Ngonga Ngomo

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 selbständig 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.

Angewandte Statistik mit R			
Applied Statistics with R			
Modulnummer: M.079.2420	**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.079.05621 Angewandte Statistik mit R	V3 Ü2	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**
keine

3	**Teilnahmevoraussetzungen:** keine *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%

<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></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 v5 (IBA v5)</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

Databases and Information Systems

Modulnummer:	**Workload (h):**	**Leistungspunkte:**	**Turnus:**
M.079.2412	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.079.05532 Databases and Information Systems	V3 Ü2	75	105	P	100/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Databases and Information Systems:
Empfohlene Vorkenntnisse
Studierende sollten Vorkenntnisse in relationalen Datenbanken und SQL haben, die in etwa denen des Moduls *Datenbanken* entsprechen, sowie Vorkenntnisse im Programmieren, die in etwa denen der Module *Programmierung 1* und *Programmierung 2* entsprechen.

4	**Inhalte:** *Inhalte der Lehrveranstaltung Databases and Information Systems:* Datenspeicherung und Datenmanagement spielen eine zentrale Rolle in verschiedensten Bereichen, z.B. Unternehmen, Forschung und Medizin, weil ein Großteil des Wissens 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, 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 • Succinct-Codierungstechniken • String-Kompressionsalgorithmen • Genom-Datenbanken • Verarbeitung riesiger Baum-Datenbestände 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, • Kompressionsverfahren für sequenzielle Daten (Strings, Bitsequenzen, ...) und Integers zu erklären, anzuwenden und zu implementieren. • Kompressionsverfahren für Bäume und Graphen und Verfahren zur Suche auf und zur Änderung der komprimierten Daten zu erklären, anzuwenden und zu implementieren. • Optimierte Suchverfahren auf Strings mit Hilfe von Suffix Array, LCP Array und Burrows-Wheeler-Transformation (BWT) sowie Verfahren zur Konstruktion von Suffix Array, LCP Array und BWT zu erklären, anzuwenden und zu implementieren. • Grammatik-basierte Kompressions- und Rekompressionsverfahren für sequenzielle Daten, Bäume und Graphen und Verfahren zur Verarbeitung der so komprimierten Daten zu erklären, anzuwenden und zu implementieren. • Herausforderungen bei der Verarbeitung von Genomdaten zu diskutieren und Verfahren zur Kompression und Sequenzierung von Genomdaten zu erklären, anzuwenden und zu implementieren. • Stärken, Schwächen und Herausforderungen der verschiedenen Kompressionsverfahren erläutern und für eine gegebene Anwendung ein geeignetes Verfahren auswählen. • sich neueste Forschungsergebnisse im Bereich dieser Lehrveranstaltung anhand von wissenschaftlichen Papers zu erarbeiten.

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Prof. Dr. Stefan Böttcher

13 **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. Zudem erwerben Studierende praktische Kenntnisse durch Computer-gestützte Übungen, in denen sie aufbauend auf den in der Vorlesung erläuterten Konzepten ihre eigenen Informationssysteme, Such- oder Kompressionsalgorithmen entwickeln und implementieren.
Lernmaterialien, Literaturangaben
Verweise auf aktuelles Lernmaterial werden in der Vorlesung gegeben.

Embedded Machine Learning

Embedded Machine Learning

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:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7	**Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		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 Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Computer Engineering v3 (CEBA v3), Bachelorstudiengang Computer Engineering v3b (CEBA v3b), Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
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.

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

1 **Modulstruktur**

	Lehrveranstaltung	Lehrform	Kontaktzeit (h)	Selbststudium (h)	Status (P/WP)	Gruppengröße (TN)
a)	L.079.05530 Foundations of the Semantic Web	V2 Ü1 P2	75	105	P	25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Foundations of the Semantic Web:
Die Vision hinter dem Semantic Web ist ein Web, in dem Menschen und Maschinen kollaborieren können, um neue Erkenntnisse effizient und datengetrieben zu gewinnen. Die derzeitige Implementierung dieser Vision ist eine Erweiterung des klassischen Webs, in der die Semantik von Daten explizit repräsentiert wird. Die wachsende Popularität und Nutzung dieses Konzepts lässt sich leicht daran ablesen, dass ca. 50% der Webseiten im Web mit Semantic Web Inhalten ausgezeichnet sind. In dieser Lehrveranstaltung werden zentrale Sprachen und Methoden des Semantic Webs vorgestellt. Die Vorlesung vermittelt alle Grundlagen, die zur erfolgreichen Implementierung und Nutzung von Semantic Web Methoden benötigt werden.
Die Lehrveranstaltung umfasst folgende Inhalte:

- Das Data Web
- Daten beschreiben: Das Resource Description Framework (RDF)
- Datenschemata: RDF Schema
- Ontologien bauen: Die Web Ontology Language (OWL)
- Daten Abfragen: SPARQL
- Datenextraktion für das Semantic Web
- Datenintegration für das Semantic Web

5 **Lernergebnisse und Kompetenzen:**

Nach Abschluss dieses Moduls können die Studierenden

- Wissensgraphen aus Spezifikationen in natürlicher Sprache modellieren,
- die Syntax und Semantik wesentlicher Sprachen aus dem Semantik Web erklären und anwenden,
- Inferenzverfahren für formale Sprachen wie RDF, RDFS und ALC erklären, implementieren und anwenden,
- Wissensgraphen aus textuellen Datenbeständen extrahieren, integrieren, säubern und fusionieren,
- Anfragen an Wissensgraphen selbständig verfassen, ausführen und auswerten.

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)

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 • Hitzler, P., Krötzsch, M., Rudolph, S., & Sure, Y. (2007). Semantic Web: Grundlagen. Springer-Verlag. • Hogan, A. et al. (2021). Knowledge graphs. ACM Computing Surveys (CSUR), 54(4), 1-37.

Introduction to Knowledge Representation and Reasoning			
Introduction to Knowledge Representation and Reasoning			
Modulnummer: M.079.2431	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 6	**Dauer (in Sem.):** 1	**Sprache:** 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:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Übungsaufgaben, schriftliche Ausarbeitung oder Referat		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 Modulabschlussprüfung bestanden ist.
10	**Gewichtung für Gesamtnote:** Das Modul wird mit der doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Computer Engineering v3 (CEBA v3), Bachelorstudiengang Computer Engineering v3b (CEBA v3b), Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
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			
Knowledge Base Engineering			
Modulnummer: M.079.2430	**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.079.05514 Knowledge Base Engineering	V3 Ü2	75	105	P	35

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

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

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

Bestehen der Studienleistung

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

Die Vergabe der Leistungspunkte erfolgt, wenn die Modulabschlussprüfung (MAP) bestanden ist.

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

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

Koordination: Prof. Dr. Yasemin Acar

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

Applied User Experience Design

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:**

keine

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.

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 oder 30-45 min	100%

<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>Fortschrittsberichte</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 doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).</td></tr>
<tr><td>11</td><td>Verwendung des Moduls in anderen Studiengängen:
Bachelorstudiengang Informatik v5 (IBA v5)</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 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.</td></tr>
</table>

Modellbasierte Softwareentwicklung			
Model-Based Software Engineering			
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.079.05503 Modellbasierte Softwareentwicklung	V3 Ü2	75	105	P	75/30

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

Teilnahmevoraussetzungen der Lehrveranstaltung Modellbasierte Softwareentwicklung:
Empfohlene Vorkenntnisse
Kenntnisse aus dem Modul *Name des Moduls bzw. der Module* sind hilfreich.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Modellbasierte Softwareentwicklung:
Inhaltsbeschreibung als Fließtext mit ca. 500-1000 Zeichen
Die Veranstaltung umfasst folgende Inhalte:

- Erstens
- . . .
- 6 bis 12 Stichpunkte

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- Lernergebnis 1
- . . .
- 4 bis 6 Lernergebnisse

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%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

12 **Modulbeauftragte/r:**

Dr. Enes Yigitbas

13 **Sonstige Hinweise:**

Hinweise der Lehrveranstaltung Modellbasierte Softwareentwicklung:
Methodische Umsetzung
Fließtext mit ca. 200-500 Zeichen
Lernmaterialien, Literaturangaben

- Literatur 1
- Literatur 2
- . . .
- Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Secure Software Engineering			
Secure Software Engineering			
Modulnummer: M.079.2115	**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.079.05640 Secure Software Engineering	V3 Ü2	75	105	P	100/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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

Im praktischen Teil “Build it, break it, fix it”, lernen Studierende, eine eigene kleine Software zu entwickeln, dann von einem anderen Team brechen zu lassen (und umgekehrt deren Implementierung zu “hacken”), und die gefundenen Schwachstellen dann zu beheben.

5	**Lernergebnisse und Kompetenzen:** Ziel der Veranstaltung ist es, dass die Studierenden 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. Nach Abschluss des Moduls können die Studierenden • die Elemente, sogenannte Security Touchpoints, eines typischen Secure Software Development Lifecycles beschreiben und unterscheiden • erklären und bewerten, inwiefern die Maßnahmen an diesen Security Touchpoints erfolgreichen Angriffen auf das zu entwickelnde System entgegenwirken • konkrete aktuelle Maßnahmen wie beispielsweise aktuelle Schnittstellen zur Authentisierung oder Verschlüsselung benennen • zudem benennen, was es zu beachten gibt, ob diese Schnittstellen korrekt eingesetzt werden • die bekanntesten Schwachstellenkategorien benennen und unterscheiden sowie erklären, welche Arten von Gegenmaßnahmen ihnen entgegenwirken • bestimmte Arten von Schwachstellen selbst in Programmcode erkennen, erklären, ausnutzen aber auch beheben.
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%

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 der doppelten Anzahl seiner Leistungspunkte gewichtet (Faktor 2).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Prof. Dr. Eric Bodden

13	**Sonstige Hinweise:** *Hinweise der Lehrveranstaltung Secure Software Engineering:* **Methodische Umsetzung** Vorlesungen und Kleingruppenübungen zur Klausurvorbereitung sowie Praktikum “Build-it, break-it, fix-it”. **Lernmaterialien, Literaturangaben:** Gary McGraw: Software Security: Building Security In (2006, Addison-Wesley Professional). Über UPB als e-book verfügbar. • Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Systematische Entwicklung von VR/AR Anwendungen			
Systematic Development of VR/AR Applications			
Modulnummer: M.079.2118	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 6	**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.079.05604 Systematische Entwicklung von VR/AR Anwendungen	V3 Ü2	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**
keine

3 **Teilnahmevoraussetzungen:**
keine

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. Die Veranstaltung umfasst folgende Inhalte: • 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.

<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, mündliche Prüfung oder Referat</td><td>90-120 min, 30-45 min oder 30 min</td><td>100%</td></tr></table></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></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 v5 (IBA v5)</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			
Systems Engineering			
Modulnummer: M.079.2119	**Workload (h):** 180	**Leistungspunkte:** 6	**Turnus:** module.frequency.de
	Studiensemester: 5	**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.079.05505 Systems Engineering	V2 Ü3	75	105	P	25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Systems Engineering:
Durch den technischen Wandel von mechatronischen zu intelligenten technischen Systemen (ITS) stehen Unternehmen und Entwicklungsteams vor vielen Herausforderungen. Wesentlich ist die Zunahme der Komplexität und Vernetzung von Systemen (Produkte). Bestehende Ansätze in der Produktentstehung können diese nicht effizient und effektiv abdecken. Systems Engineering stellt sich hierfür als vielversprechender Ansatz auf, die Herausforderungen zu lösen.
Die Veranstaltung umfasst folgende Inhalte:

- Intelligente Technische Systeme (ITS)
- Systems Engineering 101
- Der Produktlebenszyklus
- Technische & Managment Prozesse
- Tailoring und Anwendung von Systems Engineering
- Model-based Systems Engineering 101

5 **Lernergebnisse und Kompetenzen:**

Die Studierenden können

- Intelligente technische Systeme und ihre Systemelemente bestimmen und abstrahieren,
- Grundlagen des Systems Engineering verstehen und an Beispielen maßgeschneidert anwenden,
- Technische und Management Prozesse verstehen und an Beispielen anwenden,
- Grundlagen des Model-based System Engineering verstehen.

<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, mündliche Prüfung oder Referat</td><td>90-120 min, 30-45 min oder 30 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:</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 Computer Engineering v4 (CEBA v4), Bachelorstudiengang Informatik v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Roman Dumitrescu</td></tr>
<tr><td>13</td><td>Sonstige Hinweise:
Hinweise der Lehrveranstaltung Systems Engineering:
Methodische Umsetzung
• Themenspezifische Informationsvermittlung aus Theorie und Praxis
• Tieferlegung der vermittelten Inhalte über Übungen und anwendungsnahe Aufgaben
Lernmaterialien, Literaturangaben
• Gausemeier, J.; Dumitrescu, R.; Steffen, D.; Czaja, A.; Wiederkehr, O.; Tschirner, C.: Systems Engineering in industrial practice. Heinz Nixdorf Institute, University Paderborn, 2013, Unter: https://www.hni.uni-paderborn.de/en/spe/systemsengineering/
• Dumitrescu, R.; Albers, A.; Riedel, O.; Stark, R.;Gausemeier, J. (Eds): Engineering in Germany – Status quo in Business and Science. Federal Ministry of Education and Research, 2021 Unter: https://www.advanced-systems-engineering.de/#studie
• Weitere Literatur wird in der Veranstaltung bekanntgegeben.</td></tr>
</table>

User Interface Design			
User Interface Design			
Modulnummer: M.079.2130	**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.079.05499 User Interface Design	V2 Ü3	75	105	P	50/25

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

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, mündliche Prüfung oder Referat	90-120 min, 30-45 min oder 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

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

10 **Gewichtung für Gesamtnote:**

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

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

Bachelorstudiengang Informatik v5 (IBA v5)

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:**

Proseminar			
Proseminar			
Modulnummer: M.079.2530	**Workload (h):** 120	**Leistungspunkte:** 4	**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.079.075.. Proseminar	PS2	30	90	P	15

2 **Wahlmöglichkeiten innerhalb des Moduls:**

keine

3 **Teilnahmevoraussetzungen:**

keine

4 **Inhalte:**

Inhalte der Lehrveranstaltung Proseminar:
Im Proseminar wird die Einarbeitung in ein wissenschaftliches Thema erlernt und wissenschaftliches Denken und Arbeiten gestärkt. Die Inhalte werden schriftlich und mündlich präsentiert. Dazu werden Basiswissen in Bezug auf Literaturrecherche, Rhetorik und aktuelle Präsentationstechniken sowie in Bezug auf Kritikfähigkeit und Feedbackmethoden erworben und angewendet.
Die Studierenden lernen in der praktischen Durchführung das Erarbeiten eines Themas, die Auswahl von Quellen und Inhalten, 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 (Abschlussarbeit, ggf. Master-Studiengang) und im Beruf (Präsentationen, Berichte) vor. Die fachliche Ausrichtung des Proseminars ist abhängig vom jeweiligen Thema der Veranstaltung.

<table>
<tr><td>5</td><td>Lernergebnisse und Kompetenzen:
Die Studierenden können
• ein fachliches Themengebiet auf wissenschaftlichem Niveau selbständig erarbeiten, insbesondere auch durch gründliche Literaturrecherche, die das Studium wissenschaftlicher Veröffentlichungen einschließt,
• Konzepte und Sachverhalte erklären, die für das gewählte Themengebiet relevant sind,
• geeignete Quellen auswählen,
• Inhalte aus verschiedenen Quellen sichten, verstehen und zu einem Gesamtbild zusammenfügen,
• erarbeitete Erkenntnisse in Form von Ausarbeitungen im wissenschaftlichen Stil und in Form einer wissenschaftlichen Präsentation darstellen,
• fachliche Themen mit anderen diskutieren,
• die eigenen Arbeitsweisen reflektieren und kommunizieren.</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>Seminarvortrag mit schriftlicher Ausarbeitung</td><td>30-45 min; max. 20 DIN-A4-Seiten</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:</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 v5 (IBA v5)</td></tr>
<tr><td>12</td><td>Modulbeauftragte/r:
Prof. Dr.-Ing. Juraj Somorovsky</td></tr>
</table>

13	**Sonstige Hinweise:** Für das Proseminar können alle Proseminare aus dem Angebot des Bachelorstudiengangs Informatik gewählt werden. Die Anzahl der Teilnehmenden ist in der Regel auf 15 beschränkt. *Hinweise der Lehrveranstaltung Proseminar:* **Methodische Umsetzung** In den ersten semesterbegleitenden Sitzungen des Proseminars lernen die Studierenden zunächst, das Thema einzugrenzen, um dann eine wissenschaftliche Perspektive auf das zu behandelnde Thema auszuwählen und zu entwickeln. Die Studierenden werden in die wesentlichen Techniken des wissenschaftlichen Arbeitens eingeführt (Recherche, Präsentation, wissenschaftlicher Stil, Entwicklung eines roten Fadens, Umgang mit Quellen). Die Studierenden erhalten während der Erarbeitung des Themas semesterbegleitend Rückmeldungen sowohl von den Betreuenden als auch von anderen Seminarteilnehmenden. Das erarbeitete Thema wird im Rahmen des Proseminars vorgestellt. Die Präsentationen finden nach Maßgabe der Lehrenden entweder in semesterbegleitenden Sitzungen oder als Blockseminar statt. An die Präsentation schließt sich eine Diskussion an. Der Termin für die Abgabe der schriftlichen Ausarbeitung wird frühzeitig bekanntgegeben. **Lernmaterialien, Literaturangaben** Je nach gewähltem Thema.

Bachelor-Abschlussarbeit			
Bachelor Thesis			
Modulnummer: A.079.2010	**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:**

Es müssen mindestens 110 LP im Studiengang erbracht worden sein. Der Abschluss des Moduls „Proseminar“ vor Beginn der Bachelorarbeit ist zu empfehlen.

4 **Inhalte:**

Inhalte der Lehrveranstaltung Bachelor-Abschlussarbeit – Arbeitsplan:
Nach Themenabsprache mit dem Betreuer bzw. der Betreuerin erfolgt eine erste grobe Einarbeitung. Auf dieser Grundlage und einer ersten Literaturrecherche ist durch den Studenten bzw. die Studentin ein Arbeitsplan vorzulegen, der die zu erzielenden Ergebnisse samt Meilensteinen für die Arbeit dokumentiert. Die Beschreibung der zu bearbeitenden Aufgabe und Zielsetzung soll einen Umfang von in der Regel nicht mehr als fünf (in Ausnahmen bis zu zehn) Seiten haben und nach Ausgabe des Themas innerhalb eines Monats beim Betreuer bzw. der Betreuerin eingereicht werden. Der Antrittsvortrag soll zumindest die geplante Arbeit, das geplante Vorgehen und die erhofften Ergebnisse vorstellen und eine Dauer von ca. 20 Minuten haben. An den Antrittsvortrag schließt sich eine Aussprache an.

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. Die Bachelorarbeit soll einen Umfang von 50 Seiten nicht überschreiten.
Die Ergebnisse der Bachelorarbeit sind in einer Abschlusspräsentation mit anschließender wissenschaftlicher Aussprache vorzustellen. Diese findet nach Abgabe der schriftlichen Arbeit statt (in der Regel spätestens nach vier Wochen) und soll eine Dauer von ca. 30 Minuten haben. Die Abschlusspräsentation kann neben Vortrag und Aussprache weitere Elemente (z.B. die Demonstration von Software) beinhalten. Die Abschlusspräsentation geht in die Endnote mit ein.

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.
Durch die Bachelorarbeit wird nicht nur das Fachwissen in einem spezifischen Themengebiet der Informatik vertieft, es werden vor allem auch wichtige "Soft Skills" eingeübt, die für die spätere berufliche Praxis essentiell sind. Die Studierenden können

- eine Aufgabenstellung aus dem Bereich der Informatik eigenständig unter fachlicher und methodischer Betreuung mit wissenschaftlichen Methoden bearbeiten,
- die selbständige Bearbeitung einer umfangreichen Aufgabenstellung planen und durchführen,
- Anforderungen und Rahmenbedingungen einer umfangreichen Aufgabenstellung mit Betreuenden klären,
- eigene Kreativität zur Problemlösung einsetzen,
- selbständig und zielgerichtet Fachwissen und Methoden erarbeiten, um Teilprobleme zu lösen,
- die Ergebnisse in Form einer wissenschaftlichen Arbeit schriftlich und mündlich präsentieren.

6 **Prüfungsleistung:**

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

zu	**Prüfungsform**	**Dauer bzw. Umfang**	**Gewichtung für die Modulnote**
a) - b)	Bachelorarbeit mit Abschlusspräsentation	bis zu 50 Seiten bzw. ca. 30 min	100%

7 **Studienleistung, qualifizierte Teilnahme:**

zu	**Form**	**Dauer bzw. Umfang**	**SL / QT**
a)	Arbeitsplan und Antrittsvortrag	bis zu 5 Seiten bzw. 10-20 min	QT
b)	keine		

Vom jeweiligen Lehrenden wird spätestens bei der Ausgabe des Themas bekannt gegeben, wie die qualifizierte Teilnahme konkret zu erbringen ist.

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

Der Nachweis der Qualifizierten Teilnahme ist Voraussetzung für die Modulabschlussprüfung.

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

Die Vergabe von Credits erfolgt, wenn die Bachelorarbeit einschließlich des Abschlussvortrages mit fachlicher Diskussion bestanden wurde.

10	**Gewichtung für Gesamtnote:** Das Modul wird mit 60 Leistungspunkten gewichtet (Faktor 4).
11	**Verwendung des Moduls in anderen Studiengängen:** Bachelorstudiengang Informatik v5 (IBA v5)
12	**Modulbeauftragte/r:** Prof. Dr.-Ing. Juraj Somorovsky
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 50 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.

4 Module im Wintersemester

- M.079.2232 3D Modeling 67
- M.079.2420 Angewandte Statistik mit R 110
- M.079.2217 Ausgewählte Themen im Gebiet Algorithmen und Komplexität 91
- M.079.2317 Ausgewählte Themen im Gebiet Computersysteme 108
- M.079.2417 Ausgewählte Themen im Gebiet Daten und Wissen 128
- M.079.2117 Ausgewählte Themen im Gebiet Softwaretechnik 147
- A.079.2010 Bachelor-Abschlussarbeit 152
- M.079.2412 Databases and Information Systems 113
- M.079.1222 Datenstrukturen und Algorithmen 38
- M.079.1422 Einführung in Data Science 61
- M.079.1521 Einführung in die Informatik 4
- M.079.2210 Einführung in Kryptographie 73
- M.079.2211 Grundlegende Algorithmen 79
- M.079.2430 Knowledge Base Engineering 125
- M.079.2212 Komplexitätstheorie 82
- M.105.9610 Lineare Algebra für Informatik 13
- M.079.2114 Modellbasierte Softwareentwicklung 133
- M.079.1221 Modellierung 10
- M.079.1124 Programmiersprachen 35
- M.079.1121 Programmierung 1 7
- M.079.2530 Proseminar 149
- M.079.2230 Quantum Networks 85
- M.079.1322 Rechnerarchitektur 41
- M.079.1323 Rechnernetze 64
- M.079.2115 Secure Software Engineering 135
- M.079.1123 Software Engineering 31
- M.105.96300 Stochastik für Informatik 44
- Studium Generale – Bachelor 29
- M.079.2130 User Interface Design 144
- M.079.2313 Verteilte Systeme 105

5 Module im Sommersemester

- M.079.2231 3D Rendering 70
- M.105.9620 Analysis für Informatik 26
- M.079.2131 Applied User Experience Design 130
- M.079.2217 Ausgewählte Themen im Gebiet Algorithmen und Komplexität 91
- M.079.2317 Ausgewählte Themen im Gebiet Computersysteme 108
- M.079.2417 Ausgewählte Themen im Gebiet Daten und Wissen 128
- M.079.2117 Ausgewählte Themen im Gebiet Softwaretechnik 147
- A.079.2010 Bachelor-Abschlussarbeit 152
- M.079.1223 Berechenbarkeit und Komplexität 50
- M.079.1324 Betriebssysteme 53
- M.079.1421 Datenbanken 19
- M.079.1321 Digitaltechnik 23
- M.079.2318 Einführung in Hochleistungsrechnen 93
- M.048.11113 Einführung in Planungs- und Entscheidungsfindungsalgorithmen 96
- M.079.2311 Eingebettete Systeme 99
- M.079.2331 Embedded Machine Learning 116
- M.079.2414 Foundations of the Semantic Web 119
- M.079.2215 Grundlagen der Algorithmischen Geometrie 76
- M.079.1522 Informatik und Gesellschaft 58
- M.079.2431 Introduction to Knowledge Representation and Reasoning 122
- M.079.1325 IT-Sicherheit 55
- M.079.1122 Programmierung 2 16
- M.079.1125 Softwaretechnikpraktikum 47
- Studium Generale – Bachelor 29
- M.079.2118 Systematische Entwicklung von VR/AR Anwendungen 138
- M.079.2119 Systems Engineering 142
- M.079.2214 Verteilte Algorithmen und Datenstrukturen 88

6 Modules in English

- M.079.2131 Applied User Experience Design 130
- M.079.2412 Databases and Information Systems 113
- M.079.2313 Distributed Systems 105
- M.079.2331 Embedded Machine Learning 116
- M.079.2414 Foundations of the Semantic Web 119
- M.079.2211 Fundamental Algorithms 79
- M.079.2210 Introduction to Cryptography 73
- M.079.2431 Introduction to Knowledge Representation and Reasoning 122
- M.079.2430 Knowledge Base Engineering 125
- M.079.2230 Quantum Networks 85
- M.079.2115 Secure Software Engineering 135
- M.079.2217 Selected Topics in Algorithms and Complexity 91
- M.079.2317 Selected Topics in Computer Systems (BA) 108
- M.079.2417 Selected Topics in Data and Knowledge (BA) 128
- M.079.2117 Selected Topics in Software Engineering (BA) 147
- M.079.2130 User Interface Design 144

Erzeugt am 30. September 2026 um 15:16.