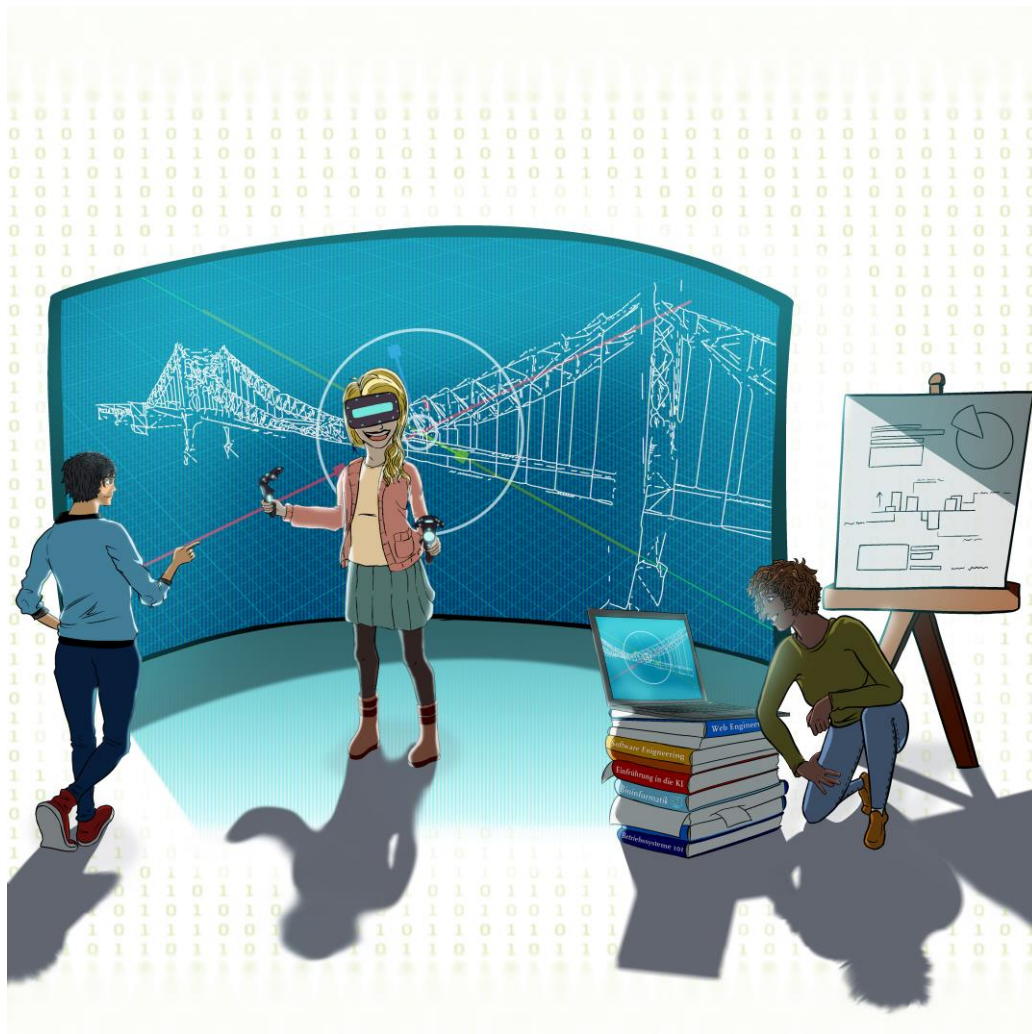


# Modulhandbuch Bachelor of Science (B.Sc.)

## Angewandte Informatik [P022]

Stand: Wintersemester 2024/25

<https://informatik.rub.de/studium/studiengaenge/ai/bsc/>



## Studienplan Bachelor Angewandte Informatik PO 22

Nr	Modul	Umfang bzw. Mind. Umfang Modul (CP)	Empfohlenes Semester	Bewertung
<b>Pflichtbereich</b>				
1	Höhere Mathematik 1	9	1	benotet
2	Informatik I	12	1	benotet
3	Technische Informatik 1	5	1	benotet
4	Höhere Mathematik 2	9	2	benotet
5	Logik	5	2	benotet
6	Informatik 2	8	2	benotet
7	Computernetze	5	2	benotet
8	Programmierung und Programmiersprachen	6	2	benotet
9	Statistik und Wahrscheinlichkeitsrechnung	5	3	benotet
10	Informatik 3	8	3	benotet
11	Software-Engineering	5	3	benotet
12	Projektmanagement	5	3	benotet
13	Wirtschaftlichkeitsanalyse	5	3	benotet
14	Einführung in die künstliche Intelligenz	5	4	benotet
15	Datenbanksysteme	7	4	benotet
16	Web-Engineering	5	4	benotet
17	Betriebssysteme	5	4	benotet
18	Software Engineering Praktikum	5	4	benotet
19	Datenschutz	5	5	benotet
<b>Wahlpflichtbereich</b>				
20	Vertiefungspraktikum*	3	4	unbenotet
22	Vertiefungsseminar*	3	5	benotet
21	Studienprojekt**	8	5	benotet
23	Vertiefungsmodule***	23	5-6	benotet
<b>Wahlbereich</b>				
24	Freie Wahlmodule	9	1-6	unbenotet
<b>Bachelorarbeit</b>				
25	Abschlussarbeit	12+3	6	benotet
Summe:		180		

#

- \* Informationen zu den im Semester wählbaren Vertiefungspraktika und Vertiefungsseminaren befinden sich im Vorlesungsverzeichnis.
- \*\* Die wählbaren Studienprojekte werden jeweils zum Ende eines Semesters fürs Folgesemester vorgestellt.
- \*\*\* Hier müssen Vertiefungsmodule aus Anwendungsbereichen der Informatik im Umfang von mindestens 23 LP gewählt werden. Informationen zu den wählbaren Modulen befinden sich im jeweils aktuellen Modulhandbuch.
- \*\*\*\* Hier können (nahezu) alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der Universitätsallianz Ruhr gewählt werden.

## Angebote Vertiefungsmodulen

Lehrveranstaltung	Lehrinheit	Umfang (CP)	Semester	Bewertung
<b>Vertiefungsmodulen</b>				
<b>Ingenieurinformatik</b>				
Geometrische Modellierung und Visualisierung	Baulng	6	WS	benotet
Grundlagen der Automatisierungstechnik	MB	5	WS	benotet
Numerische Mathematik	Mathe	5	WS	benotet
Virtuelle Produktmodellierung und -visualisierung	MB	5	WS	benotet
Digitaltechnik	Informatik	5	SS	benotet
Fertigungsautomatisierung	MB	5	SS	benotet
Menschenzentrierte Robotik	MB	6	SS	benotet
<b>Bioinformatik</b>				
Bioinformatics for Molecular and Cellular Structures (ehemals Grundlagen der Bioinformatik)	Biologie	5	WS	benotet
Methoden der Bioinformatik	Biologie	5	SS	benotet
<b>Computerlinguistik</b>				
Einführung in die Linguistik	Philologie	8	WS	benotet
Methoden der Computerlinguistik	Philologie	5	WS	benotet
Natural Language Processing with Deep Learning	Informatik	5	WS	benotet
Anwendungen der Computerlinguistik	Philologie	5 bzw. 8	Letztmalig SS 24	benotet
<b>Kryptographie und Theoretische Informatik</b>				
Einführung in die Kryptographie I	Informatik	5	WS	benotet
Einführung in die Kryptographie II	Informatik	5	SS	benotet
Highlights of Theoretical Computer Science	Informatik	10	SS	benotet
Kryptographie auf hardwarebasierten Plattformen	Informatik	5	Letztmalig WS 22/23	benotet
<b>Neuroinformatik</b>				
Artificial Neural Networks	Informatik	6	WS (nicht im WS 24/25)	benotet
Introduction to Data Science	Informatik	5	WS	benotet
Introduction to Computational Neuroscience	Informatik	6	SS	benotet
Introduction to Neural Data Science	Informatik	6	Letztmalig WS 23/24	benotet
<b>Programmier- und Simulationstechnik</b>				
Algorithmenparadigmen	Informatik	5	WS	benotet
Programming for Modern Machine Learning	Informatik	6	WS	benotet
System Performance Evaluation	Informatik	5	WS (nicht im WS 24/25)	benotet
Technische Informatik 3 - Hardware Programmierung	Informatik	5	WS	benotet
Distributed Systems	Informatik	5	SS	benotet
Functional Programming	Informatik	5	SS	benotet
Mathematics for Modelling and Data Analysis	Informatik	5	SS	benotet
Mensch-Maschine-Interaktion	AW	5	SS (nächstmalig im SS 25)	benotet
Game Development	Informatik	6	Letztmalig SS 23	benotet
Nebenläufige Programmierung	Informatik	5	Letztmalig SS 23	benotet
<b>IT in Wirtschaft, Management und Gesellschaft</b>				
Geschäftsprozess-Management	WiWi	5	WS	benotet
Wirtschaft und Digitalisierung	WiWi	5	WS	benotet
Inklusives IT-Design	AW	6	SS	benotet
Agent-based Modeling in Economics and Business	WIWI	5	Unregelmäßig (nicht im WS 24/25)	benotet

**Angebote Vertiefungsseminare und Vertiefungspraktika**

Lehrveranstaltung	Lehrinheit	Umfang (CP)	Semester	Bewertung
<b>Vertiefungsseminare</b>				
Seminar Ingenieurinformatik	Baulng	3	WS/SS	benotet
Seminar Bioinformatik	Biologie / MPC	3	WS/SS	benotet
Seminar Security Engineering	Informatik	3	WS/SS	benotet
Seminar CL (Im WS 24/25: Informationsstatus und Koreferenzauflösung (050042)	Philologie	3	WS/SS	benotet
Seminar Safety and Reliability Artificial Intelligence	Informatik	3	WS/SS	benotet
Seminar Automated Software Engineering	Informatik	3	WS	benotet
Seminar Modern Programming Languages	Informatik	3	WS	benotet
Seminar Mobile Network Security	Informatik	3	WS	benotet
Seminar Ressourceneffiziente Systemsoftware	Informatik	3	WS	benotet
Perlen der theoretischen Informatik (ehemals Grenzen in der theoretischen Informatik)	Informatik	3	WS	benotet
Seminar Distributed Systems	Informatik	3	WS	benotet
Seminar Algorithmen	Informatik	3	WS	benotet
Seminar Approximationsalgorithmen	Informatik	3	SS	benotet
Seminar Perlen der Logik (ehemals Satisfiability)	Informatik	3	SS	benotet
Seminar Computational Neuroscience	Informatik	3	SS	benotet
Seminar From Biological to Artificial Neural Networks	Informatik	3	SS	benotet
Seminar Networked Systemes	Informatik	3	SS	benotet
Seminar Reinforcement Learning	Informatik	3	SS	benotet
Seminar Implementation Security	Informatik	3	Letztmalig SS 23	benotet
Seminar Symmetrische Kryptanalyse	Informatik	3	Letztmalig WS 22/23	benotet
Seminar on Knowledge Graphs	Informatik	3	Letztmalig WS 22/23	benotet
Machine Learning Applications	Informatik	3	Letztmalig WS 23/24	benotet
Algorithms for Decision Making	Informatik	3	Letztmalig WS 23/24	benotet
Seminar on Current Topics for Systems Security and Privacy	Informatik	3	Letztmalig WS 23/24	benotet
<b>Vertiefungspraktika</b>				
Grundlagen der Roboterprogrammierung	MB	3	WS/SS	unbenotet
An Introduction to Python for Data Analysis (ehemals Introduction to Python)	Informatik	3	SS	unbenotet
Erklärbare Künstliche Intelligenz - Programmierpraktikum	AW	3	SS	unbenotet
Open Neural Data	Informatik	3	SS	unbenotet
Python programming and basic machine learning	Informatik	3	SS	unbenotet
Advanced Python Programming	Informatik	3	WS	unbenotet
Unreal C++ Praktikum	Informatik	3	WS	unbenotet
Lab Course: Challenging Problems in Reinforcement Learning	Informatik	3	WS	unbenotet
Autonomous Driving Simulation Lab	Informatik	3	WS (nicht im WS 24/25)	unbenotet
Systemsoftwaretechnik Praktikum	Informatik	3	WS (nicht im WS 24/25)	unbenotet
Praktikum Rust	Informatik	3	Letztmalig WS 23/24	unbenotet

## Abkürzungen:

AW: Institut für Arbeitswissenschaft  
 Baulng: Fakultät für Bau- und Umweltingenieurwissenschaften  
 ETIT: Fakultät für Elektrotechnik und Informationstechnik  
 MB: Fakultät für Maschinenbau  
 WIWI: Fakultät für Wirtschaftswissenschaft  
 MPC: Medizinisches Proteom Center

SS: Sommersemester  
 WS: Wintersemester  
 CP: Creditpoints

#

# MODULHANDBUCH

## Übersicht der Module

### Angewandte Informatik - Bachelor (1-Fach, PO 2022)

---

#### **Pflichtbereich**

Höhere Mathematik 1  
Wirtschaftlichkeitsanalyse  
Informatik 1  
Technische Informatik 1  
Höhere Mathematik 2  
Informatik 2  
Computernetze  
Programmierung und Programmiersprachen  
Logik  
Statistik und Wahrscheinlichkeitsrechnung  
Informatik 3  
Software Engineering  
Projektmanagement  
Einführung in die künstliche Intelligenz  
Datenbanksysteme  
Web-Engineering  
Betriebssysteme  
Software Engineering Praktikum  
Datenschutz

#### **Wahlpflichtbereich**

Introduction to Data Science  
Agent-based Modeling in Economics and Business [kein Angebot im WS 24/25]  
Algorithmenparadigmen  
Anwendungen der Computerlinguistik [B.Sc.]  
Bioinformatics for Molecular and Cellular Structures  
Digitaltechnik  
Distributed Systems  
Einführung in die Kryptographie 1  
Einführung in die Kryptographie 2  
Einführung in die Linguistik  
Fertigungsautomatisierung

Functional Programming  
Game Development  
Geometrische Modellierung und Visualisierung  
Geschäftsprozess-Management  
Grundlagen der Automatisierungstechnik  
Highlights of Theoretical Computer Science [B.Sc.]  
Inklusives IT-Design  
Introduction to Computational Neuroscience  
Künstliche Neuronale Netze (kein Angebot im WS 24/25)  
Mathematics for Modeling and Data Analysis  
Menschenzentrierte Robotik  
Mensch-Maschine-Interaktion (nächstmalig SS 25)  
Methoden der Bioinformatik  
Methoden der Computerlinguistik  
Natural language processing with deep learning [B.Sc.]  
Nebenläufige Programmierung (letztmalig SS 23)  
Numerische Mathematik  
Programming for Modern Machine Learning  
System Performance Evaluation (keine Angebot im WS 24/25)  
Technische Informatik 3  
Virtuelle Produktmodellierung und -visualisierung  
Wirtschaft und Digitalisierung  
Studienprojekt  
Vertiefungspraktikum (Angewandte Informatik)  
Vertiefungsseminar (Angewandte Informatik)

## **Wahlbereich**

Freie Wahlmodule

## **Bachelorarbeit**

Abschlussarbeit (Bachelor AI)

<b>Titel des Moduls: Höhere Mathematik 1</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> 1	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Vorlesung: Höhere Mathematik 1 (150160) Übung: Höhere Mathematik 1 (150161)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> 120 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: PD. Dr. Daniela Kacso Lehrende: PD. Dr. Daniela Kacso					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Mathematische Schulausbildung (Gymnasiale Oberstufe) und die Teilnahme an einem Mathematik-Vorkurs					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls:  Die Studierende <ul style="list-style-type: none"> <li>• Verstehen und wenden grundlegende mathematische Begriffe und Notationen an</li> <li>• Nutzen und führen die vermittelten mathematischen Methoden aus</li> <li>• Übertragen und wenden zugehörige Lösungsverfahren auch auf praktische Probleme an</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Aussagenlogik,</li> <li>• Mengen und Abbildungen</li> <li>• Reelle Zahlen und algebraische Strukturen</li> <li>• Komplexe Zahlen</li> <li>• Folgen und Reihen</li> <li>• Stetige Funktionen</li> <li>• Differenzialrechnung (in <math>\mathbb{R}</math>)</li> <li>• Integralrechnung (in <math>\mathbb{R}</math>)</li> </ul>					
<b>Lehrformen</b> Vortrag der Lehrenden in der Vorlesung (mit zum Teil digitalen Lehrformaten), Gruppenarbeit in den Übungen, Ergänzung der Bearbeitung der Hausaufgaben in Einzel- oder Gruppenarbeit durch digitale Aufgaben, Online-Tests.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestehen der schriftlichen Abschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 9/168: B.Sc. Angewandte Informatik [PO 22]					





<b>Titel des Moduls: Wirtschaftlichkeitsanalyse</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Vorlesung mit Übung (076000) Optionales Kolloquium (076001)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> 120 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: <a href="#">Dr. Barbara Wischermann</a> Lehrende: Dr. Barbara Wischermann Dr. Roland Düsing					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Keine					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls: <ul style="list-style-type: none"> <li>• sollen die Studierenden Grundbegriffe der Wirtschaftlichkeitsanalyse kennen</li> <li>• sollen die Studierenden die verschiedenen Teilgebiete der Wirtschaftlichkeitsanalyse auseinanderhalten können</li> <li>• sollen die Studierenden Aufgaben der Kostenarten- , Kostenstellen- und Kostenträgerrechnung lösen können</li> <li>• sollen die Studierenden Erfolgsgrößen errechnen und beurteilen können</li> <li>• sollen die Studierenden Methoden der Investitionsrechnung sicher anwenden</li> <li>• sollen die Studierenden kompetent mit wirtschaftlichen Fachbegriffen umgehen können</li> <li>• sollen die Studierenden in der Lage sein, sich mit Kaufleuten inhaltlich über die entsprechenden Themen auszutauschen</li> <li>• sollen die Studierenden bei Budgetverantwortung und eigenen Projekten die Grundbegriffe der Wirtschaftlichkeit kennen und berücksichtigen können</li> </ul>					
<b>Inhalt</b> <p>Angewandte Informatiker werden sich im Rahmen ihrer Berufstätigkeit als Budgetverantwortliche oder im Rahmen eines Projektmanagements auch regelmäßig mit Frage der Wirtschaftlichkeit ihres Handelns auseinandersetzen haben. Darüber hinaus werden im Kundengespräch und bei der Auftragsabwicklung Kenntnisse von Wirtschaftlichkeitsgrößen und Vorteilhaftigkeitsrechnungen als Verkaufsargumente notwendig sein. Nicht zuletzt wird sich für viele Informatiker sowohl im Rahmen eigener Investitionsüberlegungen als auch bei der Entwicklung von Software-Lösungen für Kunden die Frage nach der Wirtschaftlichkeit von Investitionsalternativen stellen. Die Lehrveranstaltung „Wirtschaftlichkeitsanalyse“ wird die Studierenden mit den Grundlagen der Wirtschaftlichkeitsanalyse vertraut machen und ihnen Methoden vermitteln, mit denen sie derartige Fragestellungen beantworten können.</p> <p>Gliederung:</p> <ol style="list-style-type: none"> <li>1. Anwendung der Wirtschaftlichkeitsanalyse (u.a. Budgetverantwortung u. Projektmanagement, Kundengespräch und Auftragsabwicklung, Investitionsentscheidungen)</li> <li>2. Grundbegriffe und begriffliche Abgrenzung</li> </ol>					

3. Die Wirtschaftlichkeit in der Leistungserstellung (u.a. Kostenarten-, Kostenstellen-, Kostenträgerrechnung; Erlöse; Erfolgsermittlung)
4. Verfahren der Wirtschaftlichkeitsrechnung in der Investitionsrechnung (statische und dynamische Verfahren)

**Lehrformen**

Vorlesung mit integrierter Übung

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (60 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

<b>Titel des Moduls: Informatik 1</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 12 CP	<b>Workload</b> 360 h	<b>Semester</b> 1	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Vorlesung und Übung: Informatik 1 (212004)  Python-Praktikum (212400)			<b>Kontaktzeit</b> 6 Semesterwochenstunden + 10 Tage (80h)	<b>Selbststudium</b> 190 h	<b>Gruppengröße</b> 400 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: <a href="#">Prof. Dr. Tobias Glasmachers</a> Lehrende: Prof. Dr. Tobias Glasmachers					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik					
<b>Vorkenntnisse</b> Keine					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen die Teilnehmer die wichtigsten Konzepte imperativer und objektorientierter Programmierung,</li> <li>• können die Teilnehmer eigene Programme entwerfen und implementieren,</li> <li>• können die Teilnehmer mit Grundbegriffen der Informatik wie etwa Korrektheit, Laufzeit, Boole'scher Algebra, Invarianten und abstrakten Datentypen arbeiten,</li> <li>• können die Teilnehmer die einfache Datenstrukturen (Arrays, Dictionaries) gezielt einsetzen und kennen Standardalgorithmen darauf, insbesondere zum Sortieren von Arrays.</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Imperative Programmierung (Variablen, Kontrollstrukturen, Funktionen und Rekursion, Fehlerbehandlung, Ereignisbehandlung)</li> <li>• einfache Datenstrukturen (Array und Dictionary)</li> <li>• Objektorientierung (Klassen, Sichtbarkeit, Schnittstellen, Vererbung)</li> <li>• Einführung in eine Reihe von Informatik-Konzepten (Invarianten, Laufzeitanalyse, Sortieralgorithmen, Repräsentation von Daten im Rechner, Boole'sche Algebra)</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Übung plus zweiwöchiges Blockpraktikum Die Vorlesung nutzt das Flipped-Classroom Lehrformat. Sämtliches Vorlesungsmaterial steht online zur Verfügung.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (150 Minuten) Erfolgreiche Teilnahme am Praktikum					
<b>Voraussetzungen für die Vergabe von Credits</b> Teilnahme am Python-Praktikum und bestandene schriftliche Modulabschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  12/168: B.Sc. Angewandte Informatik					



**Titel des Moduls: Technische Informatik 1**  
**Technical Computer Science 1**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 Stunden	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Technische Informatik 1 - Rechnerarchitektur (212013)			<b>Kontaktzeit</b> 4 SWS	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Yuval Yarom Lehrende: Prof. Yuval Yarom					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik  B.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> Keine					
<b>Lernziele (learning outcomes)</b> Upon successful completion of the course, students will be able <ul style="list-style-type: none"> <li>• Identify the main components of a modern processor, describe their functionality, and demonstrate how they each contribute to program execution</li> <li>• Evaluate computer system based on their design, and assess their impact on program performance</li> <li>• Design and develop simple programs in assembly language</li> </ul>					
<b>Inhalt</b> The course introduces the structure and function of modern computers. It explores the various components that comprise the computer, including the execution pipeline, the memory subsystem, data storage, and external devices.  A main focus of the course is exploring and analyzing program execution. This start from practical experience with assembly programming and develops to in-depth analysis of how the processor interprets and performs a program. In particular, the course identifies trade-offs made in processor design and their impact on performance.					
<b>Lehrformen</b> Contact teaching consists of two hours of lecture per week, supplemented by two hours of practical exercise sessions. Beyond contact hours, students are expected to read textbook chapters and to answer take-home exercises.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten). Bis zu 10% Bonus für Hausaufgaben.					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

<b>Titel des Moduls: Höhere Mathematik 2</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> 1	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Vorlesung (150162) Übung (150163)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> 150 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: PD. Dr. Daniela Kacso Lehrende: PD. Dr. Daniela Kacso					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Mathematische Schulausbildung (Gymnasiale Oberstufe) und die Teilnahme an einem Mathematik-Vorkurs					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls:  Die Studierenden <ul style="list-style-type: none"> <li>• Verstehen und wenden grundlegende mathematische Begriffe und Notationen an</li> <li>• Sind in der Lage die Themen zu erläutern und Einsatzmöglichkeiten zu benennen</li> <li>• Können Formalismen und Verfahren auswählen und ausführen sowie die erzielten Ergebnisse interpretieren</li> <li>• Übertragen und wenden zugehörige Lösungsverfahren auch auf praktische Probleme an</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Potenzreihen und Fourierreihen</li> <li>• Vektorräume</li> <li>• Lineare Gleichungssysteme</li> <li>• Matrizen</li> <li>• Determinanten</li> <li>• Eigenwerte und Eigenvektoren</li> <li>• Differenzialrechnung (in <math>\mathbb{R}^n</math>)</li> <li>• Gewöhnliche Differenzialgleichungen</li> </ul>					
<b>Lehrformen</b> Vortrag der Lehrenden in der Vorlesung (mit zum Teil digitalen Lehrformaten), Gruppenarbeit in den Übungen, Ergänzung der Bearbeitung der Hausaufgaben in Einzel- oder Gruppenarbeit durch digitale Aufgaben, Online-Tests					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestehen der schriftlichen Abschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					





<b>Titel des Moduls: Informatik 2</b> Computer Science 2					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> 2	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Informatik 2 - Algorithmen und Datenstrukturen (211002)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>  keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: <a href="#">Prof. Dr. Maike Buchin</a> Lehrende: Prof. Maike Buchin					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> Inhalte der Module Informatik 1 und Mathematik 1 bzw. Höhere Mathematik 1, insbesondere Programmieren und lineare Algebra					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls: <ul style="list-style-type: none"> <li>• können Studierende Algorithmen formal beschreiben und deren Korrektheit beweisen</li> <li>• können Studierende die Laufzeit und den Speicherbedarf von Algorithmen und Datenstrukturen analysieren und bewerten</li> <li>• kennen Studierende grundlegende Datenstrukturen</li> <li>• kennen Studierende grundlegende Schemata zum Entwurf von Algorithmen sind Studierende in der Lage, Algorithmen und Datenstrukturen für spezifische Probleme zu entwickeln</li> <li>• haben die Studierenden die Grundlagen der Programmiersprache Python kennengelernt</li> </ul>					
<b>Inhalt</b> Die Vorlesung gibt einen systematischen Überblick über den Entwurf und die Analyse von Algorithmen und Datenstrukturen. Dazu werden zunächst grundlegende Methoden der Analyse (insbesondere Korrektheit, Laufzeit und Speicherbedarf) von Algorithmen vorgestellt. Anschließend werden einige Algorithmen zum Sortieren und Suchen analysiert. Ebenfalls werden verschiedene grundlegende Datenstrukturen (Listen, Felder, Suchbäume und Heaps) vorgestellt. Schließlich werden Graphen betrachtet, und zwar ihre Darstellung und diverse Algorithmen auf Graphen (Durchläufe, kürzeste Wege, minimale Spannbäume). In den Übungen lernen die Studierenden sowohl die theoretische Analyse von Algorithmen und Datenstrukturen als auch deren praktische Umsetzung in eine moderne Programmiersprache (z.B. Python).					
<b>Lehrformen</b>  Hörsaalvorlesung mit Medienunterstützung und theoretische sowie praktische Übungen am Rechner					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 150 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					

8/158: B.Sc. Informatik [PO 22]

8/165: B.Sc. Informatik [PO 20]

8/168: B.Sc. Angewandte Informatik [PO 22]

8/170: B.Sc. Angewandte Informatik [PO 20]

8/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

8/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

<b>Titel des Moduls: Computernetze</b> Computer Networks					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 2	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Computernetze (211006)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 400 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr.-Ing. Christian Mainka Lehrende: Dr.-Ing. Christian Mainka					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit / Informationstechnik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende die wichtigsten Standards, die das heutige Internet verwendet.</li> <li>• kennen Studierende grundlegende Angriffskonzepte auf Computernetzwerke</li> <li>• verstehen Studierende den Zusammenhang zwischen den einzelnen Schichten eines Computernetzwerks und der darin enthaltenen Protokolle</li> <li>• können Studierende die wichtigsten Netzwerktools für Analysezwecke anwenden</li> </ul>					
<b>Inhalt</b> Die Vorlesung gibt eine Einführung in grundlegenden Protokollen und Anwendungen von Computernetzen. Der Schwerpunkt der Vorlesung liegt auf Standardprotokollen und -Algorithmen, wie sie in modernen Computernetzwerken (zum Beispiel im Internet) eingesetzt werden. Anhand eines Schichtenmodells werden die wichtigsten Grundlagen nach dem Top-Down Ansatz vorgestellt und analysiert. Dazu gehören zum Beispiel auf der obersten Schicht DNS und HTTPS im Application Layer; TCP und UDP im Transport Layer; IPv4/IPv6 und Routing Algorithmen im Network Layer; sowie MAC und ARP im untersten Link Layer. Neben der reinen Funktionsweise dieser Standards werden Sicherheitsaspekte auf allen Schichten betrachtet. Ergänzend zur Vorlesung werden Übungsaufgaben über die eLearning Plattform Moodle gestellt und in der Übungsstunde besprochen. Weiterhin wird in jeder Übung ein "Tool der Woche" vorgestellt. Dabei handelt es sich jeweils um eine spezielle Software, die man als "Netzwerker" unbedingt kennen sollte (z.B. traceroute, nmap, ...). Alle besprochenen Tools sind frei verfügbar und werden den Studenten als eine Lernplattform (virtuelle Maschine) zur Verfügung gestellt. Als Primärliteratur wird "Computernetzwerke: Der Top-Down Ansatz" von Kurose und Ross (Pearson Verlag) verwendet.					
<b>Lehrformen</b> Moodle-Unterstützte Hausaufgaben mit praxisnahen, computerunterstützten Übungen. Tool-der-Woche: Vorstellung, Einarbeitung, und Verwendung von Netzwerkrelevanten Computeranalysetools.					
<b>Prüfungsformen</b> schriftliche Modulabschlussprüfung von 120 min					

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

<b>Titel des Moduls: Programmierung und Programmiersprachen</b> Programming and programming languages					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> 2	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Programmierung und Programmiersprachen: Vorlesung und Übung (211053)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Nils Jansen Lehrende: Prof. Dr. Nils Jansen					
<b>Verwendung des Moduls</b> B. Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte des Moduls Informatik 1					
<b>Lernziele (learning outcomes)</b> Ziel dieses Kurses ist es, ein umfassendes Verständnis der wichtigsten Programmierparadigmen zu vermitteln: Imperative und objektorientierte Programmierung, funktionale Programmierung und logische Programmierung. Der Kurs stellt den Studierenden praktische und moderne Programmierübungen bereit, die ihnen bei der Lösung realer Programmierprobleme helfen. Der Schwerpunkt des Kurses liegt auf der Entwicklung, Analyse und Verifizierung von Code, der solche Probleme effizient löst. Darüber hinaus erlernen die Studierenden im Rahmen der Programmierübungen den Umgang mit modernen Softwareentwicklungstools wie IDEs und DevOps-Paketen. Nach erfolgreichem Abschluss des Kurses sind die Studierenden dazu in der Lage <ul style="list-style-type: none"> <li>• reale Programmierprobleme zur objektorientierten Programmierung zu abstrahieren</li> <li>• die Unterschiede und individuellen Vorteile der Programmierparadigmen der Objektorientierung, der funktionalen Programmierung und der logischen Programmierung verstehen</li> <li>• in JAVA zu programmieren</li> <li>• einfache Programme in Haskell und Prolog zu schreiben</li> <li>• Moderne IDEs und DevOps-Pakete zu nutzen</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Grundlagen der Programmierung in JAVA</li> <li>• Objektorientierte Programmierung in JAVA</li> <li>• GUI-Programmierung in JAVA</li> <li>• Entwurfsmuster in JAVA</li> <li>• Grundlagen der funktionalen Programmierung in Haskell</li> <li>• Grundlagen der Logikprogrammierung in Prolog</li> </ul>					
<b>Lehrformen</b> Vorlesungen, digitale Übungen, und Übungsstunden					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 6/158: B.Sc. Informatik					



<b>Titel des Moduls: Logik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Logik: Vorlesung und Übung (212013)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Thomas Zeume					
<b>Verwendung des Moduls</b>  B.Sc. Angewandte Informatik  B.Sc. Informatik					
<b>Vorkenntnisse</b> Mathematik Grundlagenvorlesungen					
<b>Lernziele (learning outcomes)</b> In dieser Veranstaltung werden die formalen Grundlagen von modernen Logiken behandelt, mit einem Fokus auf ihrer Anwendung in der Informatik. Neben der klassischen Aussagenlogik und Prädikatenlogik betrachten wir auch Modallogik. Für jede dieser Logiken formalisieren wir Syntax und Semantik, lernen wie sich informatische Szenarien in ihnen modellieren lassen, und betrachten Algorithmen und Kalküle für Unerfüllbarkeit und Folgerungsbeziehung.					
<b>Inhalt</b> Logische Methoden spielen in vielen modernen Anwendungen der Informatik eine wichtige Rolle. Aus Datenbanken werden relevante Informationen mit Hilfe auf Logik basierender Anfragesprachen extrahiert; die formale Verifikation von Software und Hardware basiert auf logischen Spezifikationssprachen und Algorithmen für diese; und Methoden für das automatisierte Schlussfolgern in der künstlichen Intelligenz haben ihre Grundlage in der formalen Logik.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  5/168: B.Sc. Angewandte Informatik [PO 22]					

<b>Titel des Moduls: Statistik und Wahrscheinlichkeitsrechnung</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Höhere Mathematik C: Vorlesung (150104) Höhere Mathematik C: Übung (150105)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 400 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Herold Dehling Lehrende: Prof. Dr. Herold Dehling					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik [PO 22]					
<b>Vorkenntnisse</b> Inhalte aus den Modulen Höhere Mathematik 1 und 2					
<b>Lernziele (learning outcomes)</b> Die Studierenden <ul style="list-style-type: none"> <li>• kennen die Grundbegriffe der Wahrscheinlichkeitstheorie und Statistik,</li> <li>• sind in der Lage, Standardaufgaben nachzuvollziehen und selbstständig zu bearbeiten,</li> <li>• kennen das Auftreten und die Bedeutung des Zufalls in Natur und Technik und</li> <li>• sind im Stande, Zufallsphänomene mit Standardverfahren zu modellieren,</li> <li>• können das Erlernte auf konkrete ingenieurwissenschaftliche Problemstellungen anwenden</li> </ul>					
<b>Inhalt</b> Die Lehrveranstaltung behandelt das zum Verständnis und zur Modellierung von Zufallsphänomenen in den Ingenieurwissenschaften erforderliche Basiswissen der Wahrscheinlichkeitstheorie und der Statistik. Hierzu gehören im Bereich der Wahrscheinlichkeitstheorie: Modellierung von Zufallsexperimenten, Wahrscheinlichkeitsraum, bedingte Wahrscheinlichkeiten, Unabhängigkeit, diskrete und stetige Zufallsvariablen, Dichte- und Verteilungsfunktion, wichtige Wahrscheinlichkeitsverteilungen (u.a. binomial, Poisson, geometrisch, normal, exponentiell, Chi-Quadrat, F-Verteilung), Erwartungswert, Varianz, Kovarianz, Korrelationskoeffizient, gemeinsame Verteilung, Faltungsformel, sowie im Bereich der Statistik: Verfahren der beschreibenden Statistik, statistische Modellierung, Grundlagen der Schätztheorie (u.a. Maximum Likelihood Methode), Konfidenzintervalle, Grundlagen der Testtheorie, Fehler 1. und 2. Art, Niveau eines Tests, Tests bei normalverteilten Stichproben (t-Test, F-Test), Lineare Regressionsmodelle (Kleinste Quadrate Methode, t-Test), Chi-Quadratstest bei diskreten Daten, 1-Faktor ANOVA.  Die Konzepte und Verfahren werden stets durch Anwendungsbeispiele und Simulationen mit Hilfe des statistischen Pakets R illustriert.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (90 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					



**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**  
5/168: B.Sc. Angewandte Informatik [PO 22]

<b>Titel des Moduls: Informatik 3</b> Computer Science 3					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Informatik 3 - Theoretische Informatik (212002)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Thomas Zeume					
<b>Verwendung des Moduls</b>  B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• beherrschen die Studierenden den professionellen Umgang mit Berechnungsmodellen und ihren Beziehungen zu Sprachklassen. Dazu gehört die intellektuelle und methodische Fähigkeit, den Nachweis der Zugehörigkeit bzw. Nichtzugehörigkeit zu einer solchen Klasse zu führen.</li> <li>• ist durch Einüben von Beweistechniken wie wechselseitige Simulation oder berechenbare Reduktionen bei den Studierenden die Einsicht gereift, dass an der Oberfläche verschieden aussehende Konzepte im Kern identisch sein können. Zudem erlaubt dies den Studierenden, neue Anwendungsprobleme selbstständig zu klassifizieren.</li> <li>• haben die Studierenden mit der Turingmaschine ein einfach handhabbares Rechnermodell erlernt, das ihnen fortan als Abstraktion für alle möglichen Rechner dient.</li> <li>• haben die Studierenden fundamentale Einsichten erlangt, welche Probleme mithilfe von Rechnern effizient entschieden, zum Teil entschieden oder prinzipiell nicht entschieden werden können. Dadurch erlangen Sie ein tieferes Verständnis der Komplexität von Berechnungsproblemen.</li> </ul>					
<b>Inhalt</b> Die Lehrveranstaltung gibt einen systematischen Überblick über die folgenden Themengebiete: <ul style="list-style-type: none"> <li>• Endliche Automaten und reguläre Ausdrücke</li> <li>• Kellerautomaten und kontextfreie Grammatiken</li> <li>• Turingmaschinen und Entscheidbarkeit</li> <li>• Nichtdeterminismus und NP-Vollständigkeitstheorie</li> </ul>					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung und Übungen, bei denen die vorgestellten Konzepte und Techniken praktisch umgesetzt werden, teilweise mit Rechnerübungen.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (180 Minuten)					

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

8/165: B.Sc. Informatik [PO 22]

8/158: B.Sc. Informatik [PO 22]

8/168: B.Sc. Angewandte Informatik [PO 22]

8/170: B.Sc. Angewandte Informatik [PO 20]

8/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

8/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

<b>Titel des Moduls: Software Engineering</b> Software Engineering					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 212000: Software Engineering			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 350 Studierende
<b>Unterrichtssprache</b> English			<b>Teilnahmevoraussetzungen</b> Programming and Programming Languages, Informatik 1		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
<b>Verwendung des Moduls</b>					
<b>Vorkenntnisse</b> Imperative and Object-Oriented Programming in a statically typed programming language Theoretical Computer Science					
<b>Lernziele (learning outcomes)</b>					
<b>Knowledge and understanding</b>					
<ul style="list-style-type: none"> <li>• Explain in detail the core activities for engineering software, including requirements engineering, software architectures and design, implementation, quality assurance (esp. testing), software process</li> <li>• explain underlying representation of programs (AST) and the compilation process</li> <li>• explain programming paradigms, such as imperative programming, generic programming, object-oriented programming, AI engineering</li> </ul>					
<b>Competence and skills</b>					
<ul style="list-style-type: none"> <li>• elicit and define software requirements in different formalisms (e.g., textual requirements, state machine diagrams, or other behavior diagrams)</li> <li>• define a software architecture upon quality requirements, using architectural patterns or styles</li> <li>• implement software architectures, frameworks, and software modules/components</li> <li>• define a software engineering process based on process models, including plan-based and agile models</li> <li>• perform quality assurance, including designing test cases according to test-case design methods for black-box and white-box testing</li> <li>• create behavioral and structural models in the context of software engineering</li> </ul>					
<b>Judgement and approach</b>					
<ul style="list-style-type: none"> <li>• identify use-cases and the potential of different SE methods and technologies for a given domain/problem</li> <li>• select and justify SE methods and technologies for a given domain/problem</li> </ul>					
<b>Inhalt</b>					
<p>You can write code, but can you also write software? The course provides an introduction into Software Engineering methods and tools. Covering the overall phases of software engineering, namely planning, requirements engineering, architectural design, implementation, quality assurance, and evolution and maintenance, the curriculum will walk students through modern software engineering technology that aims at building the modern software systems and products. The course attempts to be close to programming technology, while covering multiple paradigms and solutions.</p> <p>We cover:</p> <ul style="list-style-type: none"> <li>• Introduction to software engineering, motivated by software and project failures, laws of software engineering, and optionally a guest lecture from industry</li> </ul>					

- Introduction into compiler construction and the underlying representation of programs (e.g., abstract syntax trees, control-flow diagrams)
- Paradigms (e.g., imperative programming, generic programming, object-oriented programming, AI engineering) and software engineering principles (e.g., modularity, cohesion/coupling)
- UML behavioral and structural diagrams for modeling software
- Requirements engineering
- Software architecture and architecture implementation
- Quality assurance, including black-box and white-box testing, mutation testing, combinatorial interaction testing
- Advanced topics, such as software product lines, model-driven software engineering or security engineering (e.g., threat modeling)

#### **Lehrformen**

The teaching of this course consists of different forms: lectures, interactive quizzes, group work, group supervision, and practical assignments.

#### **Prüfungsformen**

Written exam at the end of the course (120 minutes)

#### **Voraussetzungen für die Vergabe von Credits**

Group project and final exam passed

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit/Informationstechnik [PO22]

5/149: B.Sc. IT-Sicherheit/Informationstechnik [PO20]

<b>Titel des Moduls: Projektmanagement</b> Project Management					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Projektmanagement (135040)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 150 Studierende
<b>Unterrichtssprache</b> Deutsch, Lehrmaterial teils auf Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Christian Meske Lehrende: Prof. Dr. Christian Meske					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik [PO 22]					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Studierende <ul style="list-style-type: none"> <li>• verstehen die Rolle von Projektmanagement als Steuerungsinstrument im Spannungsfeld zwischen Zeit, Kosten und Qualität</li> <li>• verfügen über Grundkenntnisse und Fähigkeiten bezogen auf klassisches sowie agiles Projektmanagement</li> <li>• kennen alle relevanten Projektabläufe und -phasen sowie deren interdependenten Zusammenhänge</li> <li>• können theoretische Lehrinhalte (im Rahmen von Übungen) direkt anwenden</li> <li>• haben ihre Fähigkeit zur Teamarbeit verbessert</li> </ul>					
<b>Inhalt</b> Die Studierenden beschäftigen sich mit Methoden und Werkzeugen des Projektmanagements in klassischen und agilen Projektszenarien. Zu den Inhalten zählen u.a.: <ul style="list-style-type: none"> <li>• Phasen eines Projekts</li> <li>• Projektmanagementprozess und Prozesselemente</li> <li>• Projekte aufsetzen und strukturieren</li> <li>• Rollen im Projekt, Projektführung</li> <li>• Elemente der Projektplanung und der Terminplanung</li> <li>• Qualitätsbegriff, Qualitätsmanagement in Projekten</li> <li>• Stakeholdermanagement</li> <li>• Risikomanagement</li> <li>• Projektumsetzung, Projektsteuerung und -controlling</li> <li>• Entscheidungen in Projekten</li> <li>• Projektabschluss</li> </ul>					
<b>Lehrformen</b> Die Veranstaltung findet über das Semester hinweg überwiegend in Präsenz statt und besteht sowohl aus einem Vorlesungs- als auch Übungsteil. Es werden klassische Vorlesungseinheiten stattfinden, ergänzt um asynchrone Online-Videos und weitere digitale Lehrinhalte. Parallel zur Vorlesung wird eine Übung angeboten (teils in Präsenz, teils virtuell) in der Studierende beispielhafte Aufgaben erhalten und bearbeiten. Zudem sind je nach Verfügbarkeit einzelne Gastvorträge aus der Praxis eingeplant.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (60 Minuten)					

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

<b>Titel des Moduls: Einführung in die künstliche Intelligenz</b> Introduction to Artificial Intelligence					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Introduction to Artificial Intelligence (211045)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 250 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Sen Cheng Lehrende: Prof. Dr. Laurenz Wiskott, Prof. Dr. Tobias Glasmachers, Prof. Dr. Sen Cheng, Prof. Dr. Gregor Schöner, Prof. Dr. Nils Jansen Prof. Dr. Setareh Maghsudi Prof. Dr. Christian Straßer					
<b>Verwendung des Moduls</b> B.Sc. Informatik (Pflichtmodul)  B.Sc. Angewandte Informatik (Pflichtmodul)  B.Sc. IT-Sicherheit/Informationstechnik (Wahlpflichtmodul)					
<b>Vorkenntnisse</b> Basic knowledge of calculus and linear algebra.					
<b>Lernziele (learning outcomes)</b> After successful completion of this course, students will be able to <ul style="list-style-type: none"> <li>• summarize a number of fundamental methods in artificial intelligence,</li> <li>• explain their mathematical basis and algorithmic nature,</li> <li>• apply them to simple problems,</li> <li>• decide which methods are suitable for which problems, and</li> <li>• communicate about the all that in English.</li> </ul>					
<b>Inhalt</b> This course gives an overview over representative methods in artificial intelligence: formal logic and reasoning, classical methods of AI, probabilistic reasoning, machine learning, deep neural networks, computational neuroscience, neural dynamics, perception, natural language processing, robotics.					
<b>Lehrformen</b>					
<b>Prüfungsformen</b> Written module final exam (120 minutes)					
<b>Voraussetzungen für die Vergabe von Credits</b> passed written exam					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/158: B.Sc. Informatik [PO 22]					



5/170: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/Informationstechnik [PO 20]

<b>Titel des Moduls: Datenbanksysteme</b> Database Systems					
<b>Modul-Nr./Code</b>	<b>Credits</b> 7 CP	<b>Workload</b> 210 h	<b>Semester</b> 4	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Datenbanksysteme (211008)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> 250 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Veelasha Moonsamy Lehrende: Dr. Veelasha Moonsamy					
<b>Verwendung des Moduls</b> B.Sc. Informatik [PO 22]  B.Sc. Angewandte Informatik [PO 22]					
<b>Vorkenntnisse</b> Grundkenntnisse der Informatik (Inhalte der Module Informatik 1 – Programmierung, Technische Informatik 1 – Rechnerarchitektur)					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Absolvieren des Moduls <ul style="list-style-type: none"> <li>• erlangen die Studierenden ein Grundverständnis von modernen Datenbanksystemen, ihrer Funktion und ihrer Implementierung</li> <li>• haben die Studierenden Datenmodellierungstechniken erlernt</li> <li>• haben die Studierenden die Semantik und die Syntax des Entity-Relationships Modells kennengelernt</li> <li>• kennen die Studenten das relational Datenbankmodell und die Relationale Algebra</li> <li>• kennen die Studierenden Anfragesprachen (z.B. SQL) und können diese nutzen</li> <li>• verstehen die Studierenden die Konzepte von Transaktion und Fehlerbehandlung</li> <li>• haben die Studierenden unterschiedliche Datenbankmanagementsysteme kennengelernt</li> <li>• sind die Studierenden in der Lage, neue Datenbanken zu modellieren und zu implementieren</li> <li>• haben die Studenten Kenntnisse über die Prozesse hinter einer Datenbankabfrage und wie diese optimiert werden kann</li> </ul>					
<b>Inhalt</b> Die Datenbanktechnologie ist eine Schlüsseltechnologie der praktischen und angewandten Informatik. Zentrales Thema dieser Veranstaltung sind die Modellierung, Aufbau und die Nutzung von Datenbanken. Folgende Themen werden behandelt: <ul style="list-style-type: none"> <li>• Einführung in Datenbanksysteme</li> <li>• Entity-Relationship Modell und Verbesserungen</li> <li>• das relational Datenbankmodell</li> <li>• Relationale Algebra und Kalkül</li> <li>• Die Relationale Anfragesprache SQL</li> <li>• Datenbankprogrammierung</li> <li>• physische Datenorganisation</li> <li>• Anfragebearbeitung und Optimierung</li> <li>• Transaktionsverwaltung und Fehlerbehandlung</li> </ul>					
<b>Lehrformen</b> In der wöchentlichen Vorlesung werden die Lerninhalte theoretisch vermittelt. In der unterstützenden wöchentlichen Übung werden theoretische Fragestellungen sowie praktische Fragestellungen und Aufgaben am Computer bearbeitet. Die Aufgaben und Lösungen werden in der Übung gemeinsam erarbeitet und besprochen.					

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

7/158: B.Sc. Informatik [PO 22]

7/168: B.Sc. Angewandte Informatik [PO 22]

<b>Titel des Moduls: Web-Engineering</b> Web-Engineering					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> Siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Web-Engineering (128968 + 128969)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 200 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Markus König Lehrende: Prof. Dr.-Ing. Markus König Stephan Embers, M.Sc.					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik  M.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> Programmieren, Architektur von Web-basierten Systemen					
<b>Lernziele (learning outcomes)</b> Die Entwicklung von Web-Anwendungen und Web-Services ist zentraler Bestandteil der Digitalisierung. Ziel der Lehrveranstaltung ist die Vermittlung von Grundlagen und bewährten Verfahren in der Web-Entwicklung. Studierende lernen konzeptuelle technologische Bausteine kennen: Transportverfahren, Webseitendarstellung, dynamische Web-Anwendungen und Web-Services. Über das konzeptuelle Verständnis hinaus werden praktische Kompetenzen vermittelt. Dazu werden moderne Werkzeuge der Web-Entwicklung, sowohl server- als auch clientseitig, vorgestellt und in den Übungssitzungen praktisch vertieft. Während der Umsetzung einfacher Web-Anwendungen stehen auch analytische Fähigkeiten im Fokus: Studierende werden befähigt, verschiedene Verfahren in Hinblick auf Performanz und Wartbarkeit zu bewerten. Diese Fähigkeiten sind in der kritischen Planungsphase von Software-Projekten unerlässlich. Nach dem erfolgreichen Abschluss des Moduls: <ul style="list-style-type: none"> <li>• kennen Studierende gängige Konzepte der Web-Entwicklung in den Aspekten Präsentation, Transport und Bereitstellung von Daten</li> <li>• beherrschen Studierende grundlegende Fähigkeiten in Webseitendarstellung, dynamischen Web-Anwendungen und modernen Services (Node.js)</li> </ul>					
<b>Inhalt</b> Im Rahmen des Moduls werden den Studierenden aktuelle Techniken und Kenntnisse im Bereich der Web-Entwicklung aufgezeigt. Thematisch wird der Bereich der server- und clientseitigen Entwicklung abgedeckt. JavaScript stellt dabei eine zentrale Rolle dar. Folgende Lehrinhalte werden behandelt: <ul style="list-style-type: none"> <li>• Einführung in clientseitige Web-Entwicklung: HTML, CSS, JavaScript, Web Components</li> <li>• Transportverfahren und deren Nutzung: Representational State Transfer (REST), Asynchrones JavaScript und XML (AJAX)</li> <li>• Serverseitige Entwicklung mit Node.js und weiterführende Technologien</li> </ul>					
<b>Lehrformen</b> Synchrone Onlinevorlesung, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend					

bereitgestellten Materialien und Aufgaben.

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

<b>Titel des Moduls: Betriebssysteme</b> Operating Systems					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 4	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Betriebssysteme (211005)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 350 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>  keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Timo Hönig Lehrende: Prof. Dr.-Ing. Timo Hönig					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit/Informationstechnik					
<b>Vorkenntnisse</b> Grundkenntnisse der Informatik (Inhalte der Module Informatik 1 – Programmierung und Technische Informatik 1 – Rechnerarchitektur)					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Absolvieren des Moduls <ul style="list-style-type: none"> <li>• erlangen die Studierenden ein solides Grundverständnis von modernen Betriebssystemen, ihrer Funktion und ihrer Implementierung</li> <li>• sind die Studierenden in der Lage, verschiedene Aspekte eines Betriebssystems wie Prozess- und Speichermanagement zu verstehen und zu nutzen, sie können dabei verschiedene Designentscheidungen eigenständig analysieren und bewerten</li> <li>• sind die Studierenden in der Lage, bestimmte Aspekte eines Betriebssystems selbst zu designen und diese argumentativ zu verteidigen</li> </ul>					
<b>Inhalt</b> In diesem Modul werden die wichtigsten Grundlagen zu Betriebssystemen vorgestellt. Dazu gehören zum Beispiel: <ul style="list-style-type: none"> <li>• Betriebssystemkonzepte</li> <li>• Prozesse und Threads, Interprozesskommunikation</li> <li>• Scheduling-Mechanismen</li> <li>• Speicherverwaltung, Speicherabstraktionen, Paging</li> <li>• Dateisysteme</li> <li>• Eingabe- und Ausgabeverwaltung</li> <li>• Algorithmen zur Vermeidung von Deadlocks</li> <li>• Grundlagen der Sicherheit von Betriebssystemen</li> </ul> <p>In den letzten Wochen der Veranstaltung, abhängig vom verfügbaren Zeitfenster, werden spezielle Themen wie beispielsweise Multimedia-Betriebssysteme, Multiprozessorsysteme und Entwurf von Betriebssystemen, behandelt.</p> <p>Um den Bezug zu modernen Betriebssystemen (aktuellen Versionen von Linux, Windows und macOS) herzustellen, werden die Themen an praktischen Beispielen illustriert. Dies ermöglicht es den Studierenden, die in der Vorlesung besprochenen Themen praktisch nachzuvollziehen.</p>					

**Lehrformen**

Die Vorlesung wird als seminaristischer Unterricht mit Medienunterstützung abgehalten. eLearning unterstützte Hausaufgaben mit praxisnahen, am Rechner zu implementierenden Übungen werden alle zwei Wochen vergeben und in der Übungsstunde besprochen.

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (90 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/158: B.Sc. Informatik [PO 22]

5/170: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/Informationstechnik [PO 20]

<b>Titel des Moduls: Software Engineering Praktikum</b> Software Engineering Lab					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Software Engineering Lab (211500)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 5 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Programmiererfahrung in einer objektorientierten Programmiersprache (am besten Java), sicherer Umgang mit git					
<b>Lernziele (learning outcomes)</b> Nach dem Modulabschluss <ul style="list-style-type: none"> <li>• können die Studierenden agil arbeiten nach SCRUM</li> <li>• wissen Studierende, wie man kleine Software-Projekte plant und können diese in Java-Android umsetzen</li> <li>• können Studierende ihre eigenen Ergebnisse in angemessener Form präsentieren</li> </ul>					
<b>Inhalt</b>  Im Software Engineering Lab wird in kleinen Projektgruppen eine Android App mit AndroidStudio entwickelt. In der begleitenden Vorlesung werden die Grundlagen moderner Softwareentwicklung vermittelt und im Projekt praktisch umgesetzt. Die Projektgruppen arbeiten selbstorganisiert agil und werden durch den gesamten Entwicklungsprozess unterstützend angeleitet. Die Entwicklung beginnt mit der Backlog-Erstellung und endet mit einem kurzen Produkt-Pitch.					
<b>Lehrformen</b> Agiles Arbeiten in Projektgruppen					
<b>Prüfungsformen</b> Projektarbeit (semesterbegleitend) mit Zwischenmeetings, Abgaben und Abschlusspräsentation					
<b>Voraussetzungen für die Vergabe von Credits</b>  Erfolgreich abgeschlossene Projektarbeit mit Abschlusspräsentation.  Zum Bestehen des Moduls ist die Teilnahme am SCRUM-Workshop, an allen Supervisions- und Kundenterminen sowie an der Abschlusspräsentation (Prüfung) formal verpflichtend. Wenn Sie nicht teilnehmen können, informieren Sie bitte vorher Ihren Betreuer und Ihre Gruppenmitglieder und legen Sie die erforderlichen Unterlagen (z. B. ärztliches Attest bei Krankheit) vor. Außerdem müssen Sie, wie in der ersten Vorlesung erläutert, eine Gruppe bilden und diese rechtzeitig bei uns anmelden. Die Anwesenheit bei den Vorlesungen ist dringend erwünscht, aber keine Pflicht.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					



5/158: B.Sc. Informatik [PO 22]

5/168: B.Sc. Angewandte Informatik [PO 22]

<b>Titel des Moduls: Datenschutz</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Datenschutz (260081)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> 120 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Kai-Uwe Loser Lehrende: Dr. Kai-Uwe Loser					
<b>Verwendung des Moduls</b> B. Sc. Angewandte Informatik  B.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b>  Datenschutz befasst sich mit der Frage, wie man Bürger, Arbeitnehmer, Kunden, Patienten etc. vor negativen Auswirkungen durch die Verarbeitung von Daten zu ihrer Person schützen kann. Es besteht die Anforderung an Informatiker, Computersysteme so zu gestalten, dass sie die Umsetzung datenschutzrechtlicher Prinzipien unterstützen. Die Vorlesung befasst sich daher mit den Prinzipien des Datenschutzrechtes und den praktischen Auswirkungen für Informatiker. Dabei wird vor allem Wert daraufgelegt, diese zentralen Prinzipien verstehbar zu machen. Neben dem Datenschutzgrundverordnung werden auch Spezialregelungen behandelt, die z.B. für die Regulierung der Telekommunikation, oder für den Einsatz elektronischer Datenverarbeitung in der Arbeitswelt zum Einsatz kommen. Die DSGVO ist inzwischen auch über den europäischen Raum hinaus ein akzeptierter Standard. Unterschiedliche Rechtsphilosophische Betrachtungen werden thematisiert, um zu vermitteln, wo international Sichtweisen und Fragestellungen divergieren. Insgesamt wird das Thema konstruktiv betrachtet: das Thema Privacy by Design, wird auf allen Ebenen betrachtet.  Lernziel der Vorlesung ist es, dass die Studierenden künftig in der Lage sind, zu erkennen, an welchen Stellen ihres beruflichen Wirkens der Datenschutz relevant ist, und wie sie vorgehen müssen, um sich geeignete Informationen oder Sachverstand zu besorgen. Das zu vermittelnde Wissen soll so grundlegend sein, dass man sich auch auf neue Entwicklungen (wie etwa Novellierungen und Ergänzungen des Bundesdatenschutzgesetzes) einstellen kann.  Nach dem erfolgreichen Abschluss des Moduls					
<ul style="list-style-type: none"> <li>• kennen Studierende die Grundzüge des Datenschutzrechtes,</li> <li>• verstehen Studierende die gesellschaftlichen Hintergründe,</li> <li>• können Datenverarbeitungsprozesse hinsichtlich der Relevanz des Datenschutzrechts analysieren und</li> <li>• können Lösungsmuster anwenden, um Systeme datenschutzfreundlich und datenschutzrechtskonform zu gestalten.</li> </ul>					
<b>Inhalt</b>					
<ul style="list-style-type: none"> <li>• Was ist Datenschutz, informationelle Selbstbestimmung und Privacy?</li> <li>• Welche Folgen haben Verarbeitungen personenbezogener Daten? Woher entstehen diese Folgen?</li> </ul>					

- Was sind die Prinzipien des Datenschutzes
- Welche Rechte haben die von der Verarbeitung betroffenen Personen?
- Was passiert mit personenbezogenen Daten in vernetzten Systemen?
- Welche organisatorischen und technischen Maßnahmen helfen, personenbezogene Daten zu sichern?
- Was ist Privacy by Design und wie kann das umgesetzt werden?
- Spezielle Bereiche der Datenverarbeitung: Telekommunikation, Wirtschaft, Medizin

**Lehrformen**

Vorlesung mit Folien, Übungen zu Wissens- und Verständnisabfragen sowie Anwendung auf Beispiele

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (90 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B. Sc. Angewandte Informatik [PO 22]

5/170: B. Sc. Angewandte Informatik [PO 20]

5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

<b>Titel des Moduls: Introduction to Data Science</b> Introduction to Data Science					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Introduction to Data Science (212039)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 100 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Robert Schmidt Prof. Dr. Bilal Zafar Lehrende: Prof. Dr. Robert Schmidt Prof. Dr. Bilal Zafar					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik					
<b>Vorkenntnisse</b> Grundkenntnisse in Infinitesimalrechnung, linearer Algebra und Programmierung in Python					
<b>Lernziele (learning outcomes)</b> Am Ende dieses Kurses werden Sie mit folgenden Themen vertraut sein <ul style="list-style-type: none"> <li>1. Die wichtigsten modernen Methoden für datengestützte Vorhersagen</li> <li>2. Methoden zur Verarbeitung, Erkundung und Visualisierung von Daten verschiedener Modalitäten wie Bild, Text und Tabellen</li> <li>3. Aufbau von Proof-of-Concept-Code für die Lösung realer Data-Science-Probleme</li> <li>4. Fragen rund um Vertrauen und mögliche Abhilfemaßnahmen bei Anwendungen von Data Science.</li> </ul>					
<b>Inhalt</b> Data Science ist ein sich schnell entwickelnder Bereich mit zahlreichen Anwendungsgebieten. In diesem Kurs lernen Sie grundlegende Werkzeuge von Data Science kennen. Sie werden auch mit fortgeschrittenen Methoden des Deep Learning und deren praktischen Anwendungen vertraut gemacht. Im ersten Teil des Kurses erhalten Sie eine Einführung in die grundlegenden statistischen Methoden, die Data Science zugrunde liegen. Sie erlernen auch Techniken zur Analyse und Visualisierung von Datensätzen unterschiedlicher Modalitäten wie Text, Bilder und Tabellen. Sie werden tief in die datengesteuerten Vorhersagemethoden des maschinellen Lernens und des Deep Learning eintauchen. Im letzten Teil des Kurses führen wir Sie in fortgeschrittene Themen ein, einschließlich der jüngsten Fortschritte bei der Modellierung großer Sprachen und der Nutzung datengesteuerter Entscheidungsfindung auf vertrauenswürdige Weise.					
<b>Lehrformen</b> Jede Sitzung des Kurses besteht aus einem Vorlesungsteil, in dem die theoretischen Konzepte vorgestellt werden, und einem praktischen Teil, in dem Sie praktische Erfahrungen mit Jupyter Notebooks sammeln können.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					

5/168: B.Sc. Angewandte Informatik

5/158: B.Sc. Informatik

**Titel des Moduls: Agent-based Modeling in Economics and Business [kein Angebot im WS 24/25]****Agent-based Modeling in Economics and Business [no offer in WS 24/25]**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> unregelmäßig, angeboten im SS 23	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Agent-based modeling in Economics and Business (075261)			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Michael Roos Lehrende: Prof. Dr. Michael Roos					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Gute Englischkenntnisse, Grundlegende Programmierkenntnisse					
<b>Lernziele (learning outcomes)</b> Das Modul verfolgt das Ziel, Studierende der angewandten Informatik und Studierende der Wirtschaftswissenschaft in die Methode der agenten-basierten Modellierung und Simulationstechniken einzuführen. Hierbei sollen Studierende relevante volkswirtschaftliche Fragestellungen mit quantitativen Ansätzen bearbeiten. Im Rahmen des Moduls erwerben Studierende folgende Kenntnisse: <ul style="list-style-type: none"><li>• Überführung von wissenschaftlichen Fragestellungen aus dem Bereich Volkswirtschaftslehre in agenten-basierte Computersimulationen</li><li>• Praktische Arbeit mit agenten-basierten Modellen, Interpretation von Simulationsergebnissen und mögliche Limitationen des Ansatzes</li><li>• Grundlegendes Wissen, um eigene Modelle zu implementieren und Simulationen selbständig durchzuführen</li><li>• Die Programmiersprache NetLogo</li></ul>					
<b>Inhalt</b> In der Vorlesung wird die Methode der agentenbasierten Computersimulation vorgestellt und gezeigt, wie sie zur Analyse komplexer ökonomischer Systeme angewendet werden kann. Die Methode der agenten-basierten Computersimulationen gewinnt in der Forschung sowie in der Praxis weiter an Relevanz. Dazu werden Beispiele aus der betriebswirtschaftlichen und der volkswirtschaftlichen Forschungsliteratur präsentiert. Während der Seminartermine werden verschiedene Modelle vorgestellt, die dann als Grundlage für Gruppendiskussionen und eigene Programmieraufgaben dienen. Hier steht die praktische Anwendung im Vordergrund.					
<b>Lehrformen</b> Vorlesung und					
<b>Prüfungsformen</b> Semesterbegleitend; Hausarbeit in Form eines Lab-Reports und Vortrag im Seminarkolloquium					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestehen der folgenden Leistungen: 1) Studienleistung (unbenotet) 2) Hausarbeit und Vortrag (benotet)					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

<b>Titel des Moduls: Algorithmenparadigmen</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Algorithmenparadigmen (211043)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Maïke Buchin Lehrende: Prof. Dr. Maïke Buchin					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte aus den Modulen (Höhere) Mathematik 1 und 2 sowie aus den Modulen Informatik 1-3					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende eine Reihe von Algorithmenparadigmen</li> <li>• können Studierende basierend auf den Paradigmen effiziente Algorithmen für Probleme entwickeln</li> <li>• verstehen Studierende die Vor- und Nachteile unterschiedlicher Paradigmen</li> </ul>					
<b>Inhalt</b> In der Vorlesung betrachten wir unterschiedliche Algorithmenparadigmen, also Schemata zum Entwurf von effizienten Algorithmen. Dazu betrachten wir zunächst die bereits bekannten Paradigmen inkrementell, Teile-und-Herrsche und gierig und wenden diese auf verschiedene Probleme an. Darauf aufbauend lernen wir Dynamisches Programmieren kennen, sowie die Methoden Backtracking und Branch-and-Bound. Auch betrachten wir ein Paradigma speziell für geometrische Probleme: das Sweepline-Verfahren.					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung sowie Tutorien als seminaristischer Unterricht					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten) oder mündliche Prüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche oder mündliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/170: B.Sc. Informatik [PO 22]  5/158: B.Sc. Informatik [PO 20]  5/168: B.Sc. Angewandte Informatik [PO 22]  5/170: B.Sc. Angewandte Informatik [PO 20]					





**Titel des Moduls: Anwendungen der Computerlinguistik [B.Sc.]**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h bzw. 240 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Wechselndes Angebot (siehe im jeweiligen Semester unter Veranstaltungen)			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 120 h bzw. 210 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Stefanie Dipper Lehrende: Prof. Dr. Stefanie Dipper, Dr. Ronja Laarman-Quante, u.a.					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte aus den Bachelormodulen "Einführung in die Linguistik" sowie "Methoden der Computerlinguistik". Alternativ oder ergänzend wird das Durcharbeiten von Standard-Lehrbüchern der Computerlinguistik empfohlen, z. B. von: Jurafsky, D., Martin, J. H. (2008). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Prentice-Hall.					
<b>Lernziele (learning outcomes)</b> 5 CP-Variante  Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"><li>• verfügen Sie über spezifische Kenntnisse über ein Teilgebiet der Computerlinguistik und beginnen, eigenständiger mit computerlinguistischen Forschungsergebnissen zu arbeiten:</li><li>• Sie kennen in einem Teilgebiet der Computerlinguistik (z. B. automatische Textzusammenfassung, Dialogmodellierung, Koreferenzauflösung, Metaphernanalyse, natürlichsprachliche Generierung) die wichtigsten Methoden und können sie auf andere Daten anpassen und (in gegebenenfalls vereinfachter Form) nachimplementieren.</li><li>• Sie können die Ansätze aus computerlinguistischen Fachpublikationen verstehen, zueinander in Zusammenhang setzen und Bezüge zwischen den Studien in schriftlichen Arbeiten und in Präsentationen formulieren.</li></ul> 8 CP-Variante:  Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"><li>• verfügen Sie über vertiefte Kenntnisse in mehreren Teilgebieten der Computerlinguistik.</li><li>• Sie können eigenständig Literatur zu einer computerlinguistischen Fragestellung recherchieren, die Methoden miteinander vergleichen und sie für schriftliche Arbeiten und Präsentationen zu einem eigenen wissenschaftlichen Beitrag in Beziehung zu setzen.</li><li>• Sie können eigenständig Ansätze nachimplementieren und nach computerlinguistischen Standards evaluieren.</li></ul>					

**Inhalt**

5 CP-Variante:

In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt. Zu jedem Thema werden Sie zunächst die zentralen Methoden erarbeiten und sich einen Überblick verschaffen, welche Ansätze es in diesem Bereich gibt. Das Modul zielt darauf ab, Sie eigenständiger in der Arbeit mit computerlinguistischer Fachliteratur zu machen sowie Sie in die Lage zu versetzen, computerlinguistische Ansätze nachzuimplementieren. Sie lernen, computerlinguistische Forschungsergebnisse in Beziehung zueinander zu setzen.

8 CP-Variante:

In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt. Anders als in der 5 CP-Variante steht die eigenständige Erarbeitung des Seminarthemas im Vordergrund. Das Modul zielt darauf ab, Sie noch unabhängiger in der Arbeit mit computerlinguistischen Fachartikeln zu machen, so dass sie eigenständig kleinere Literaturüberblicke verfassen und Systeme implementieren können.

Die Modulprüfung in Form einer Implementation mit Research Thesis zu einer im Seminar entwickelten Fragestellung dient insofern als Vorbereitung für die BA-Arbeit.

**Lehrformen**

Seminaristischer Unterricht mit Gruppenarbeiten, mündlichen Präsentationen von Forschungsarbeiten.

Die Betreuung bei Selbststudiumanteilen von 150h+ pro Lehrveranstaltung erfolgt nach individueller Absprache. Im Regelfall erfolgen individuelle Sitzungen (Zoom oder Präsenz) im wöchentlichen oder 14-tägigen Wechsel.

**Prüfungsformen**

5 CP-Variante:

Die benotete Modulprüfung wird im Regelfall durch ein Research Paper (ca. 3.000 Wörter) erbracht.

8 CP-Variante:

Die benotete Modulprüfung wird im Regelfall durch eine Implementation begleitet von einer Research Thesis (ca. 6.000 Wörter) erbracht, deren Thema Sie in Rücksprache mit dem/der Seminarleiter/innen wählen.

**Voraussetzungen für die Vergabe von Credits**

Erfolgreicher Abschluss der Modulprüfung (2 bzw. 5 CP) sowie bestandene Studienleistung (3,0 CP) im Seminar.

In den zugehörigen Lehrveranstaltungen steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht in allen Grundkursen und Seminaren Anwesenheitspflicht und Sie sollten nach Möglichkeit nicht fehlen. Das bedeutet im Einzelnen:

Erlaubte Fehlzeiten in Seminaren: max. 3 Sitzungen

Über diese Sitzungen hinaus dürfen Sie nur mit Entschuldigung (z. B. mit einem ärztlichen Attest) fehlen. In der ersten Sitzung besteht grundsätzlich Anwesenheitspflicht, weil dann die grundlegenden Informationen zum Ablauf des Kurses besprochen werden. Darüber hinaus können die Lehrenden einzelne Sitzungen, die z. B. für gemeinsames Arbeiten wesentlich sind, als verpflichtend erklären.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168 bzw. 8/168: B.Sc. Angewandte Informatik [PO 22]

5/170 bzw. 8/170: B.Sc. Angewandte Informatik [PO 20]

<b>Titel des Moduls: Bioinformatics for Molecular and Cellular Structures</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>  Lecture: Bioinformatics for Molecular and Cellular Structures (190702)  Exercise for Lecture: Bioinformatics for Molecular and Cellular Structures (190703)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Axel Mosig Lehrende: Prof. Dr. Axel Mosig, Prof. Dr. Raphael Stoll, Dr. Daniel Todt					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Grundkenntnisse in (Molekularer) Biologie.					
<b>Lernziele (learning outcomes)</b> Die Teilnehmer/innen erwerben Grundkenntnisse und erhalten einen Einblick in die aktuellen Werkzeuge und zugrunde liegenden Methoden der Bioinformatik. Erworbene Kompetenzen liegen vor allem im Bereich des Erlernens bioinformatischer Werkzeuge, des Identifizierens angemessener Bioinformatik Methoden für biologische Fragestellungen sowie das Erlernen von formalem mathematisch-informatischen Denkens. Hierbei spielt das Erlernen interdisziplinären Denkens und das Anwenden von Fähigkeiten und Wissen über Fächergrenzen hinweg eine besondere Rolle.					
<b>Inhalt</b>  Bioinformatische Werkzeuge und Methoden sind zu einem festen Bestandteil der biologischen Forschung geworden. Die Vorlesung gibt einen Überblick über die aktuellen Werkzeuge und die zugrunde liegenden Methoden, mit einem besonderen Schwerpunkt der Analyse von Sequenz und Struktur von Proteinen. In den Übungen wird einerseits die Anwendung dieser Werkzeuge in der Praxis vermittelt, andererseits die theoretischen Grundlagen anhand von Übungsaufgaben vertieft.					
<b>Lehrformen</b>  Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  5/168: B.Sc. Angewandte Informatik [PO 22]					



<b>Titel des Moduls: Digitaltechnik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Technische Informatik 2 - Digitaltechnik für ITS und Informatik (211014)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch, Vorlesungs- und Übungsfolien auf Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Tim Güneysu Lehrende: Prof. Dr. Tim Güneysu					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte des Moduls Mathematik 1 – Grundlagen bzw. Höhere Mathematik 1 und 2. Vorausgesetzt wird ein generelles Interesse an technischen Systemen, die Fähigkeit zu strukturieren, algorithmischem Denken sowie die Fähigkeit zum Erfassen von komplexen Abhängigkeiten und Interaktionsmustern.					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls haben die Studierenden umfassende Kenntnisse in Boolescher Algebra, Struktur und Funktionsweise grundlegender digitaler Schaltungen, Kostenoptimierung digitaler Funktionsgruppen, Techniken zur taktsynchronen Verarbeitung von Daten, Kodierung und Verarbeitung von Daten, Struktur und Funktionsweise solcher Grundfunktionalitäten, die insbesondere in Mikroprozessorarchitekturen zentrale Bestandteile sind, erworben. Die Studierenden sind in der Lage, grundlegende Schaltungskonzepte digitaler Logik- und Funktionsblöcke zu verstehen, ihr Zusammenspiel zu analysieren, die Funktionalität zu bewerten und einfache Blöcke selbst zu entwickeln. Weiterhin werden die Bewertung und Entwicklung von mehrstufigen kombinatorischen Logikblöcken sowie von Finite State Machines (FSMs) behandelt. Die Studierenden erlernen die Hardwarebeschreibungssprache Verilog, und zu jedem Thema der Vorlesung werden Verilog-Beispiele gegeben. Die Vorlesung befasst sich ausschließlich mit (takt-)synchronen Schaltungen.					
<b>Inhalt</b> Der Kurs gibt einen systematischen Überblick über die folgenden Themen: Boolesche Algebra, Realisierung boolescher Funktionen, Minimierung boolescher Funktionen, Multiplexer, Kodierer, Dekodierer, fehlererkennende und fehlerkorrigierende Codes, Addierer, Subtrahierer, Multiplizierer, Hardwarebeschreibungssprache Verilog, Speicherelemente (Flipflops), sequentielle Schaltungen, Zähler, Schieberegister, RAM, Finite State Machines (FSMs), Timing-Analyse sequentieller Schaltungen, und kurzer Überblick über FPGAs.					
<b>Lehrformen</b> Die Vorlesung wird als seminaristischer Unterricht abgehalten, die Übungen entweder am Rechner oder mit Stift und Papier.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 120 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

**Titel des Moduls: Distributed Systems**

Distributed systems

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Distributed Systems (211004)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf  
Lehrende: Prof. Dr.-Ing. Steffen Bondorf**Verwendung des Moduls**

B.Sc. Angewandte Informatik

B.Sc. Informatik [unter dem deutschen Modultitel]

**Vorkenntnisse**

Allgemeines Interesse an technischen Systemen

**Lernziele (learning outcomes)**

Die Studierenden sollen eine breite Kenntnis über die auftretenden Herausforderungen beim Entwurf und bei der Anwendung von verteilten Computersystemen erlangen. Nach dem erfolgreichen Abschluss des Moduls

- kennen die Studierenden verschiedene Systemmodelle und Architekturen, die zum Entwurf sowie zu der Klassifizierung von verteilten Systemen dienen. Sie können verschiedene Rollen von Teilsystemen differenzieren und sie formal beschreiben
- können die Studierenden vielfältige Herausforderungen beim Aufbau eines verteilten Systems identifizieren und kennen die wichtigsten Standardtechniken zum Umgang mit diesen, inklusive deren Vor- und Nachteile
- können die Studierenden die Funktionsweise eines verteilt implementierten Systems anhand dessen Beschreibung verstehen und die ausgeführte Aufgabe herleiten - können die Studierenden die Fähigkeit eines verteilten Systems zur Erfüllung seiner Aufgabe beurteilen, die Quellen potenzieller Probleme identifizieren und können Verbesserungen sowie deren Integration entwerfen
- sind die Studierenden in der Lage, gegebene Alternativen zur verteilten Implementierung eines Systems für eine bestimmte Aufgabe zu bewerten und begründet in eine Rangfolge zu bringen

**Inhalt**

Diese Lehrveranstaltung behandelt grundlegende Architekturen und Methoden, die die Funktionsfähigkeit leistungsfähiger verteilter Computersysteme ermöglichen. Ein solches verteilte System dient der Erfüllung einer bestimmten Aufgabe unter Verwendung von mehreren unabhängigen Teilsystemen und soll dem Benutzer dabei jedoch wie ein einzelnes Computersystem erscheinen. Um dies zu erreichen, müssen die verschiedenen Teilsysteme über gemeinsames Wissen verfügen. Es treten durch die Verteilung im Vergleich zu einzelnen Systemen eine Reihe von Herausforderungen auf, die den Inhalt der Vorlesung bilden: Teilsysteme müssen sich gegenseitig auffinden können, sie müssen in der Lage sein, Nachrichten auszutauschen, Daten müssen trotz der so entstehenden Replikation über Teilsysteme hinweg konsistent gehalten werden, Fehler in Teilsystemen müssen tolerierbar sein und die Ressourcen des Gesamtsystems sollen möglichst effizient genutzt werden, sodass die gegebene Aufgabe performant erfüllt wird. All diese Komponenten und Aspekte finden sich in modernen, Internetbasierten Systemen wieder. Sie garantieren die Funktionsfähigkeit von Diensten wie das World Wide Web, E-Mail oder File-Sharing.



**Lehrformen**

Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten

**Prüfungsformen**

Schriftliche Modulabschlussprüfung über 120 Minuten

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

<b>Titel des Moduls: Einführung in die Kryptographie 1</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Einführung in die Kryptographie 1 (212010)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 300 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Christof Paar Lehrende: Prof. Dr.-Ing. Christof Paar					
<b>Verwendung des Moduls</b> B.Sc. IT-Sicherheit  B.Sc. Informatik  B.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/ Netze und Systeme					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b>  Nach erfolgreichem Abschluss des Moduls verfügen die Studierenden über Kenntnisse der grundlegenden Anwendungen symmetrischer Verfahren und über Grundkenntnisse der asymmetrischen Kryptographie. Sie können entscheiden, unter welchen Bedingungen man in der Praxis bestimmte Verfahren einsetzt und wie die Sicherheitsparameter zu wählen sind. Mit den Grundlagen des abstrakten Denkens in der IT-Sicherheitstechnik sind sie vertraut. Zum anderen erreichen die Studierenden durch Beschreibungen ausgewählter praxisrelevanter Algorithmen, wie z. B. des AES- oder RSA-Algorithmus, ein algorithmisches und technisches Verständnis zur praktischen Anwendung. Die Studierenden erhalten dabei einen Überblick über in der Praxis eingesetzten Lösungen. Sie sind in der Lage, argumentativ eine bestimmte Lösung zu verteidigen. Die Vorlesungen werden auch als Videos in Deutsch und Englisch angeboten. Die Studierenden können daher durch das zweisprachige eLearning-Angebot Sprachkompetenzen in der Wissenschaftssprache Englisch erwerben.					
<b>Inhalt</b> Das Modul bietet einen allgemeinen Einstieg in die Funktionsweise moderner Kryptographie und ihrer Bedeutung für die IT-Sicherheit. Es werden grundlegende Begriffe und mathematisch/technische Verfahren der Kryptographie erläutert. Praktisch relevante symmetrische und asymmetrische Verfahren und Algorithmen werden vorgestellt und an praxisrelevanten Beispielen erläutert.  Die Vorlesung lässt sich in zwei Teile gliedern:  Die Grundlagen der symmetrischen Kryptographie einschließlich der Beschreibung einiger historischer Verschlüsselungsverfahren (Caesar Chiffre, Affine Chiffre), aktueller symmetrischer Verfahren (AES, 3-DES) und grundlegender Konzepte wie dem One-Time-Pad und Stromchiffren werden im ersten Teil behandelt. Benötigte mathematische Grundlagen, insbesondere modulares Rechnen und endliche Körper, werden ebenfalls aus					

Anwendersicht eingeführt.

Der zweite Teil besteht aus einer Einführung in die asymmetrische Kryptographie und der Vorstellung eines ihrer wichtigsten Stellvertreter, dem RSA-Verfahren. Hierzu wird eine Einführung in die Grundlagen der Zahlentheorie durchgeführt, die für die asymmetrische Kryptoverfahren relevant sind (u. a. Ringe ganzer Zahlen und der euklidische Algorithmus).

In beiden Vorlesungsteilen werden aktuelle Sicherheitseinschätzungen und Implementierungsaspekte der vorgestellten Chiffren auch jeweils diskutiert.

#### **Lehrformen**

Vorlesung mit Übung, die Veranstaltung wird digital angeboten

#### **Prüfungsformen**

Klausurarbeit (120 Minuten)

#### **Voraussetzungen für die Vergabe von Credits**

Erfolgreiches Bestehen der Modulklausur.

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/149: B.Sc. IT-Sicherheit [PO 20]

5/150: B.Sc. IT-Sicherheit [PO 22]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/96 : M.Sc. IT-Sicherheit/ Netze und Systeme [PO20]

5/99 : M.Sc. IT-Sicherheit/ Netze und Systeme [PO22]

## **Titel des Moduls: Einführung in die Kryptographie 2**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Einführung in die Kryptographie 2 (211009)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 300 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		

### **Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr.-Ing. Christof Paar  
Lehrende: Prof. Dr.-Ing. Christof Paar

### **Verwendung des Moduls**

B.Sc. IT-Sicherheit  
B.Sc. Informatik  
B.Sc. Angewandte Informatik  
M.Sc. IT-Sicherheit/ Netze und Systeme

### **Vorkenntnisse**

Inhalte der Vorlesung "Einführung in die Kryptographie 1"

### **Lernziele (learning outcomes)**

Nach erfolgreichem Abschluss des Moduls verfügen die Studierenden über Kenntnisse der grundlegenden Anwendungen asymmetrischer und hybrider Verfahren. Sie können entscheiden, unter welchen Bedingungen man in der Praxis bestimmte Verfahren einsetzt und wie die Sicherheitsparameter zu wählen sind. Mit den Grundlagen des abstrakten Denkens in der IT-Sicherheitstechnik sind sie vertraut. Zum anderen erreichen die Studierenden durch Beschreibungen ausgewählter praxisrelevanter Algorithmen, wie z.B. des Diffie-Hellmann-Schlüsselaustausch oder ECC-basierten Verfahren, ein algorithmisches und technisches Verständnis zur praktischen Anwendung. Die Studierenden erhalten dabei einen Überblick über die in Unternehmen eingesetzten Lösungen. Sie sind in der Lage, argumentativ eine bestimmte Lösung zu verteidigen. Die Vorlesungen werden zusätzlich auch als Videos in Deutsch und Englisch angeboten. Die Studierenden können daher durch das zweisprachige eLearning-Angebot Sprachkompetenzen in der Wissenschaftssprache Englisch erwerben.

### **Inhalt**

Das Modul bietet einen allgemeinen Einstieg in die Funktionsweise moderner Kryptografie und Datensicherheit. Es werden grundlegende Begriffe und mathematisch/technische Verfahren der Kryptografie und der Datensicherheit erläutert. Praktisch relevante asymmetrische Verfahren und Algorithmen werden vorgestellt und an praxisrelevanten Beispielen erläutert. Die Vorlesung lässt sich in zwei Teile gliedern:

Der erste Teil beginnt mit einer Einleitung zu asymmetrischen Verfahren und deren wichtigsten Stellvertretern (Diffie-Hellman, elliptische Kurven). Der Schwerpunkt liegt auf der algorithmischen Einführung der asymmetrischen Verfahren, die sowohl Verschlüsselungsalgorithmen als auch digitale Signaturen beinhalten. Abgeschlossen wird dieser Teil durch Hashfunktionen, die eine große Rolle für digitalen Signaturen und Message Authentication Codes (MACs oder kryptografische Checksummen) spielen.

Im zweiten Teil der Vorlesung werden Grundlagen von Sicherheitslösungen aufbauend auf den Konzepten der symmetrischen und asymmetrischen Kryptographie besprochen. Dabei wird vor allem auf die in Unternehmen

notwendigen und eingesetzten Lösungen (PKI, digitale Zertifikate etc.) eingegangen.

**Lehrformen**

Vorlesung mit Übungen

**Prüfungsformen**

Klausurarbeit (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/150: B.Sc. IT-Sicherheit/Informationstechnik [PO 22]

5/149: B.Sc. IT-Sicherheit/Informationstechnik [PO 20]

5/165: B.Sc. Informatik [PO 22]

5/158: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/96 : M.Sc. IT-Sicherheit / Netze und Systeme [PO20]

5/99 : M.Sc. IT-Sicherheit / Netze und Systeme [PO22]

<b>Titel des Moduls: Einführung in die Linguistik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Einführung in die Linguistik (050004)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch und Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Stefanie Dipper Lehrende: Dr. Ronja Laarmann-Quante					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<p><b>Lernziele (learning outcomes)</b>  Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über grundlegendes Wissen und elementare analytische/methodische Fertigkeiten in den zentralen Teildisziplinen der Linguistik, die Sie kennenlernen – Phonetik/Phonologie, Morphologie, Syntax, Semantik, Pragmatik.  Sie können</p> <ul style="list-style-type: none"> <li>• die grundlegenden Begriffe und Konzepte der Phonetik/Phonologie, Morphologie, Syntax, Semantik und Pragmatik auflisten, dabei terminologisch korrekt benennen und Sie können die genannten Konzepte auch darstellen.</li> <li>• Die grundlegende Terminologie, der im Modul behandelten, linguistischen Teildisziplinen sowie verschiedener Schulen und zentraler Forschungsgebiete der Linguistik sind Ihnen sowohl auf Deutsch als auch auf Englisch bekannt und Sie können sprachwissenschaftliche Sachverhalte und Prozesse zutreffend bezeichnen, aufzählen und definieren.</li> <li>• Sie verstehen somit die linguistischen Fachbegriffe der o. a. Teildisziplinen (insbesondere auch in englischsprachiger Literatur, die für uns der Normalfall ist) und können sie auch richtig anwenden.</li> <li>• Sie können die elementaren analytischen Methoden der strukturellen Linguistik in Phonetik/Phonologie, Morphologie, Syntax, Semantik und Pragmatik identifizieren und die entsprechenden Prozeduren skizzieren.</li> <li>• Folglich können Sie auch elementare Darstellungen und Analysen klassifizieren und den verschiedenen linguistischen Teildisziplinen und Forschungsgebieten zuordnen.</li> <li>• Sie können erste, einfache Analysen sprachlicher Daten auf den Beschreibungsebenen der Phonetik/Phonologie, Morphologie, Syntax, Semantik und Pragmatik durchführen und können dafür Methoden je nach analytischer Aufgabenstellung richtig auswählen, zuordnen und natürlich anwenden.</li> </ul>					
<p><b>Inhalt</b>  Dieser Kurs vermittelt die grundlegenden Konzepte und Strukturmerkmale der zentralen linguistischen Teildisziplinen bzw. Beschreibungsebenen (Phonetik/Phonologie, Morphologie, Syntax, Semantik, Pragmatik):  <b>Phonetik/Phonologie:</b>  Die Phonologie untersucht lautliche Struktur/en von Sprachen. Dabei befasst sie sich u.a. mit den für ihre Beschreibung und Analyse relevanten wichtigsten physiologischen und physikalischen Gegebenheiten. In diesem Grundkurs lernen Sie zunächst phonologische Methoden und Analysen sowie grundlegendes Begriffsinventar und wesentliche Analysemethoden u. a. in den folgenden Bereichen kennen:  · Sprachlaute vs. Schrift (Transkription, IPA);  · artikulatorische Parameter, Lautklassifikation und segmentale Komposition;</p>					

- phonemische Analyse, phonologische Regeln;
- Silben und Silbifizierung.

#### Morphologie:

Die Morphologie untersucht primär die interne Struktur von Wörtern. Sie lernen morphologische Methoden und Analysen kennen sowie grundlegendes Begriffsinventar und wesentliche Analysemethoden. Es geht dabei u. a. um Betrachtung, Diskussion und Anwendung unterschiedlicher Aspekte und Methoden morphologischer Beschreibung und Analyse. Behandelt werden dabei u. a.

- Morphembegriff, Affigierung, Allomorphie;
- Eigenschaften von Derivation und Flexion in den Sprachen der Welt;
- formale Typen morphologischer Operationen.

#### Syntax:

Syntax beschäftigt sich mit der Kombination von Wörtern zu komplexeren Einheiten bis hin zu Sätzen. Sie untersucht dabei, welche Wörter mit welchen anderen kombiniert werden können, welche Stellungsregularitäten dabei auftreten und wie die resultierende Struktur mit der Bedeutung solcher Einheiten zusammenhängt. In diesem thematischen Teilbereich des Grundkurses lernen Sie grundlegende Konzepte der Syntax kennen, darunter u. a.

- Wortarten, syntaktische Kategorien und Funktionen;
- semantische Rollen, Valenz, Konstituenz, Dependenz.

#### Semantik:

Semantik ist die Teildisziplin der Linguistik, das sich mit den Bedeutungen sprachlicher Ausdrücke beschäftigt. Ein zentrales Thema ist folglich die Beantwortung der Frage, was man unter Bedeutung eines sprachlichen Ausdrucks versteht. Dabei lernen Sie grundlegende Bedeutungsarten kennen und beschäftigen sich zunächst mit den Bedeutungen von Wörtern, sprich der lexikalischen Semantik. Ferner bekommen Sie einen Überblick über Bedeutungsphänomene auf Satzebene. Die Themen dieses Grundkurses sind u. a.:

- Bedeutungsarten: deskriptive, soziale, expressive Bedeutung;
- Ausdrucks-, Äußerungsbedeutung und kommunikativer Sinn;
- Referenz und Proposition;
- semantische Relationen, Polysemie, Metonymie, Metapher.

#### Pragmatik:

Pragmatik befasst sich schwerpunktmäßig mit der Bedeutung sprachlicher Ausdrücke im (situativen) Kontext. Die Themen für diesen Teilbereich des Grundkurses sind Betrachtung, Diskussion und Anwendung von Konzepten und Methoden der Pragmatik, Sprachgebrauch und Bedeutungsaspekte der Sprachverwendung, darunter u. a.

- Interaktion zwischen Pragmatik und Semantik;
- Konversationsmaximen und Implikaturen;
- Präsuppositionen, Deixis, Sprechakte.

### **Lehrformen**

Die Lehrform ist aufgrund der Gruppengröße vermittlungsorientiert, d. h.

- Präsentation durch Kursleiter/in mit Fragen an die Teilnehmer/innen und Diskussion einzelner Aspekte, sowie
- kleinere Übungen bzw. Analysen während der Sitzungen, die Sie z. B. mit Ihren Sitznachbarn zusammen durchführen.
- Dedizierte Übungssitzungen, in denen praktische Aspekte des Kurses interaktiv eingeübt werden

### **Prüfungsformen**

Die benotete Modulabschlussprüfung wird in Form von schriftlichen Hausaufgaben in diesem Grundkurs sowie einer abschließenden Klausur voraussichtlich gegen Ende der Vorlesungszeit erbracht. Die genauen Bedingungen dieser Prüfungsleistung bzw. dieser Modulabschlussprüfung werden frühzeitig im Grundkurs bekannt gegeben.

### **Voraussetzungen für die Vergabe von Credits**

1. Bestandene Modulabschlussklausur
2. Bestandene Hausarbeiten

Wichtig:

In dieser Lehrveranstaltung steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht Anwesenheitspflicht und Sie sollten nach Möglichkeit nicht fehlen. Das bedeutet im Einzelnen:

Erlaubte Fehlzeiten im Modul Einführung in die Linguistik: max. 6 Sitzungen im Theorieteil, max. 3 Sitzungen im Praxisteil

Über diese Sitzungen hinaus dürfen Sie nur mit Entschuldigung (z. B. mit einem ärztlichen Attest) fehlen. In der ersten Sitzung besteht grundsätzlich Anwesenheitspflicht, weil dann die grundlegenden Informationen zum Ablauf des Kurses besprochen werden. Darüber hinaus können die Lehrenden einzelne Sitzungen, die z. B. für gemeinsames Arbeiten wesentlich sind, als verpflichtend erklären.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

8/168: B.Sc. Angewandte Informatik [PO 22]

8/170: B.Sc. Angewandte Informatik [PO 20]



## **Titel des Moduls: Fertigungsautomatisierung**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Fertigungsautomatisierung (136460)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Bernd Kuhlenkötter Lehrende: Dr.-Ing. Jannis Sinnemann					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Die vorherige erfolgreiche Teilnahme am Modul Grundlagen der Automatisierungstechnik ist keine Voraussetzung, aber ggf. hilfreich.					
<b>Lernziele (learning outcomes)</b> Aufbauend auf den in den Grundlagen der Automatisierungstechnik vermittelten Kenntnissen lernen die Studierenden vertiefte ingenieurwissenschaftliche Grundlagen im Bereich der automatisierten Fertigungsverfahren mit bahngesteuerten Werkzeugen kennen. Dabei sollen vier zentrale Lernziele auf Modulebene erreicht werden: <ul style="list-style-type: none"><li>• Die Studierenden verstehen den technischen Aufbau, die Einsatzmöglichkeiten sowie Vor- und Nachteile von unterschiedlichen Automatisierungskomponenten. Dieses Wissen können die Studierenden anwenden, um für einen Fertigungsprozess geeignete Automatisierungskomponenten auszuwählen.</li><li>• Für Roboter/Bewegungsautomaten können kinematische Ketten abgeleitet und die Koordinatensysteme der einzelnen Achsen dargestellt werden. Es können Koordinatentransformationen von der Basis bis zum Endeffektor abgeleitet und ein Steuerungsprogramm entworfen werden.</li><li>• Bei Greiftechnik können die Studierenden die wirkenden Kräfte und Beschleunigungen ermitteln. Zudem können Greifer in Bezug auf deren Wirkprinzipien analysiert werden.</li><li>• Für gegebene Fertigungsprobleme können die Studierenden eine Wirtschaftlichkeitsbetrachtung erstellen sowie Kosten und Risiken ableiten. Zudem können bekannte Lösungen auf andere Automatisierungsprobleme übertragen und geeignete Lösungskonzepte geplant werden.</li></ul>					
<b>Inhalt</b> In der Veranstaltung werden zunächst unterschiedliche Fertigungsverfahren (z. B. Roboter in der spanenden Fertigung, Roboter in Umformprozessen) vorgestellt. Im Fokus stehen dabei die Automatisierungsmöglichkeiten der jeweiligen Verfahren. Anhand von praxisnahen Beispielen werden die spezifischen Anforderungen an die Automatisierung herausgearbeitet. Nachfolgend werden Roboter/ Bewegungsautomaten diskutiert. Dabei wird das gesamte Spektrum vom Aufbau unterschiedlicher Getriebe, über einzelne Gelenke bis zum Aufbau von Industrierobotern behandelt und es werden Vor- und Nachteile der einzelnen Automaten, bspw. in Bezug auf deren Genauigkeit, erarbeitet. Ein weiterer Abschnitt behandelt die für Werkzeugmaschinen und Roboter wichtigen Wegmesssysteme. Mit Blick auf eine vollständige Fertigungsautomatisierung wird im Anschluss auf Materialfluss- und Ordnungseinrichtungen sowie auf Endeffektoren eingegangen. Dabei liegt ein Fokus auf der Greiftechnik, welche eine Handhabung von Werkstücken ermöglicht und gemäß der Vielfalt an Werkstücken ein großes Spektrum an unterschiedlichen Greifprinzipien und entsprechenden Umsetzungen bietet. Im Weiteren werden unterschiedliche Fertigungskonzepte und -systeme diskutiert. Dabei werden sowohl Gründe und Möglichkeiten für die Automatisierung erläutert als auch Vor- und Nachteile unterschiedlicher Fertigungslösungen aus technischer und wirtschaftlicher Sicht abgeleitet. Die einzelnen Aspekte werden anhand realer Beispiele verdeutlicht.					

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

**Titel des Moduls: Functional Programming**  
**Functional Programming**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Functional Programming (211060)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> kein		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Dr. Clara Schneidewind  
 Lehrende: Dr. Clara Schneidewind  
 Dr. Roberto Blanco  
 Dr. Catalin Hritcu

**Verwendung des Moduls**

B.Sc. Informatik  
 B.Sc. Angewandte Informatik  
 M.Sc. IT-Sicherheit / Informationstechnik

**Vorkenntnisse**

Es sind keine besonderen Vorkenntnisse erforderlich. Der Kurs ist für Studierende aller Niveaus zugänglich und dürfte sowohl für Studierende interessant sein, die bereits über Programmierkenntnisse verfügen und ein eleganteres Programmierparadigma erlernen möchten, als auch für Studierende ohne vorherige Programmiererfahrung.

**Lernziele (learning outcomes)**

Nach erfolgreichem Abschluss dieses Kurses können die Studierenden:

- Programme in funktionalen Hochsprachen, insbesondere in OCaml, entwickeln
- die Verwendung von Rekursion zur Definition von Datenstrukturen (Listen, Maps, Bäume, etc.) und rein funktionalen Algorithmen verstehen und anwenden
- den Aufbau und die Vorteile von Typensystemen verstehen und sie zur Unterstützung des Programmierentwurfs und der Implementierung einzusetzen
- fortgeschrittene Funktionen der funktionalen Programmierung analysieren, einschließlich Typ-Polymorphismus und Funktionen höherer Ordnung
- informelle Überlegungen zur Korrektheit und Effizienz funktionaler Programme anstellen und formale Alternativen zur Argumentation kennen
- Anwendung von Typabstraktion und Modularisierung zur Strukturierung von Programmen in Sammlungen von Bibliotheken und Verwendung dieser Bibliotheken, um darauf komplexere Programme aufzubauen
- die grundlegenden Prinzipien des Entwurfs von Programmiersprachen verstehen, insbesondere in Bezug auf die funktionale Programmierung
- einfache Programmiersprachen entwerfen und entwickeln, einschließlich ihrer formalen Definition und der anschließenden Implementierung als Interpreter

**Inhalt**

Dieser Kurs bietet eine fundierte und dennoch praxisnahe Einführung in die Prinzipien und Praxis der Programmierung nach dem funktionalen Paradigma. Diese zunehmend populäre Disziplin basiert auf der Definition und Ausführung von Funktionen zur Durchführung von Berechnungen. In der reinen funktionalen Programmierung, die im Mittelpunkt des Kurses steht, stellen diese Funktionen in sich abgeschlossene Berechnungen dar, die andere Teile des Programms nicht beeinflussen. Infolgedessen sind funktionale Programme elegant, effizient, leicht zu verstehen und zu ändern und schließen viele häufige Fehlerquellen aus.

die bei der imperativen Programmierung auftreten. Wir untersuchen die Prinzipien, die das Design funktionaler Sprachen leiten, wie sie funktionieren und warum. Im Mittelpunkt dieser Untersuchung steht der Begriff der starken Typisierung und der Entwurf von Typsystemen, die es uns ermöglichen, ausdrucksstarke und gut funktionierende Programme zu schreiben.

Während des gesamten Kurses werden die theoretischen Grundlagen der Programmiersprache OCaml sorgfältig von den ersten Prinzipien an erklärt und durch interaktive Übungen sofort in die Praxis umgesetzt. Diese decken alles ab, was die Studenten benötigen, um komplexe funktionale Programme zu entwickeln, einschließlich erweiterter Fallstudien wie eine kleine Programmiersprache, die auf OCaml selbst basiert. Die im Kurs erlernten Techniken ermöglichen es den Studierenden, sich zu geschickten Programmierern zu entwickeln und viele fortgeschrittene Funktionen auch in den gängigen Programmiersprachen zu nutzen, da diese zunehmend von den Fortschritten der funktionalen Programmierung beeinflusst werden. Darüber hinaus erwerben sie die notwendigen Grundlagen, um ihr Wissen über die Programmierung und deren Verbindung zur Informatik, Mathematik und Logik zu vertiefen. Da die funktionale Programmierung die primäre Alternative zur imperativen Programmierung darstellt und das direkteste Mittel zur Erstellung korrekter Programme ist, kann jeder, der ein besserer Programmierer werden möchte, von diesem Kurs profitieren.

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestehen der Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/158: B.Sc. Informatik [PO 22]

5/170: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

5/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

<b>Titel des Moduls: Game Development</b> Game Development					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Game Development (211001)			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch, Kursmaterial auf Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: M. Sc. Daniel Vonk					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Grundlegende Programmierkenntnisse; Zugang zu einem Computer mit Internet, auf dem die Unity-Engine läuft ( <a href="https://unity3d.com/de/get-unity/download">https://unity3d.com/de/get-unity/download</a> )					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verstehen die Studierenden Grundlagen der objektorientierten Programmierung mit C# im Rahmen der Unity-Engine,</li> <li>• haben die Studierenden ein umfassendes Wissen über den Bereich der Spieleentwicklung erworben und kennen moderne Tools sowie aktuelle Methoden der 2D- und 3D-Entwicklung,</li> <li>• können die Studierenden praxisnahe Problemstellungen der Softwareentwicklung analysieren und eigenständig lösen,</li> <li>• können die Studierenden Projekte im Bereich der Spieleentwicklung definieren und fachgerecht umsetzen</li> </ul>					
<b>Inhalt</b> Die Veranstaltung bietet einen umfangreichen Einblick in viele Bereiche der Spieleentwicklung. Dazu gehören: <ul style="list-style-type: none"> <li>• Grundlagenwissen (Spiele-Engines, moderne Softwaretools, Projektmanagement)</li> <li>• C#-Grundlagen (Syntax, Datentypen, Operatoren, Kontrollstrukturen)</li> <li>• Benutzerinteraktion (In-/Output mit Tastatur sowie Controller, User Interfaces)</li> <li>• Gameplay (Bewegen von Spielobjekten, Kamerasteuerung, Game Loop und Framerates)</li> <li>• Physik (Rigidbody, Collider, Trigger)</li> <li>• Assets (Import von Bildern, Audio und 3D-Modellen sowie Erstellung von Animationen)</li> <li>• Grafik (Texturen, Partikeleffekte, Beleuchtung, Post-Processing)</li> <li>• Leveldesign (Tilemaps, 3D-Umgebungen, Terrains) Studierende setzen das erlernte Wissen durch die Entwicklung einfacher Computerspiele in der Unity-Engine um. Die erworbenen Fähigkeiten lassen sich jedoch einfach auf andere Software-Frameworks übertragen.</li> </ul>					
<b>Lehrformen</b> Online-Videos und wöchentliche Hörsaalübungen					
<b>Prüfungsformen</b> Semesterbegleitende Projektarbeiten					

**Voraussetzungen für die Vergabe von Credits**

Bestandene Projektarbeiten; Studienleistung: Bearbeitung wöchentlicher Tests, die sich inhaltlich auf die einzelnen Kursabschnitte beziehen. Für die Zulassung zu den Projektarbeiten müssen die jeweiligen Tests erfolgreich absolviert worden sein.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

6/158: B.Sc. Informatik [PO 22]

6/165: B.Sc. Informatik [PO 20]

6/168: B.Sc. Angewandte Informatik [PO 22]

6/170: B.Sc. Angewandte Informatik [PO 20]

**Titel des Moduls: Geometrische Modellierung und Visualisierung**  
**Geometric Modeling and Visualization**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Geometrische Modellierung und Visualisierung (129008)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Markus König Lehrende: Prof. Dr.-Ing. Markus König					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Die Studierenden <ul style="list-style-type: none"> <li>• erarbeiten wesentliche geometrische Methoden im Ingenieurwesen und wenden diese zielgerichtet auf Ingenieuraufgaben an,</li> <li>• sind in der Lage, verschiedene geometrische Aufgabenstellungen aus Forschung und Praxis unter Verwendung aktueller Methoden der Mathematik und Informatik zu lösen,</li> <li>• können die theoretischen Grundlagen der behandelten Ansätze anwenden und Forschungsergebnisse aus diesen Bereichen reflektieren und beurteilen.</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Geometrische Modelle</li> <li>• Affine Abbildungen und Differentialgeometrie</li> <li>• Freiformkurven und Freiformflächen</li> <li>• Boundary Representation</li> <li>• Constructive Solid Geometry</li> <li>• Octrees</li> <li>• Zerlegung und Triangulierung</li> <li>• Visualisierungstechniken</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Übung am PC					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (180 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 6/168: B.Sc. Angewandte Informatik [PO 22]  6/170: B.Sc. Angewandte Informatik [PO 20]					





<b>Titel des Moduls: Geschäftsprozess-Management</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Vorlesung: Geschäftsprozess-Management (074170) Übung: Geschäftsprozess-Management (074171)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Marion Steven Lehrende: Dr. Roland Düsing					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Die TeilnehmerInnen des Kurses sollen <ul style="list-style-type: none"> <li>• die funktionsorientierte und prozessorientierte Organisation von Unternehmen unterscheiden und bewerten können,</li> <li>• die Zielsetzung, Phasen und Ausrichtungen des Geschäftsprozess-Managements erklären können,</li> <li>• die Grundsätze, Sprachen und Werkzeuge der Modellierung von Geschäftsprozessen charakterisieren können,</li> <li>• auf der Grundlage einer Modellierungssprache Geschäftsprozess- Modelle einfacher Komplexität entwickeln können, Geschäftsprozess-Modelle im Hinblick auf verschiedene Kriterien wie z. B. Prozesszeit, Prozessqualität oder Prozesskosten bewerten und verbessern können.</li> </ul>					
<b>Inhalt</b> Geschäftsprozess-Management ermöglicht eine zielgerichtete Steuerung der Geschäftsprozesse eines Unternehmens. Ein Geschäftsprozess, wie z. B. die Bearbeitung einer Kundenanfrage oder der Verkauf eines Produkts, besteht aus einer zusammenhängenden abgeschlossenen Folge von Tätigkeiten, die auf die Erfüllung der Bedürfnisse von Kunden ausgerichtet ist. Gleichzeitig sollen Geschäftsprozesse die strategischen und operativen Zielvorgaben erfüllen, von denen der Geschäftserfolg eines Unternehmens abhängt. In diesem Modul werden die Grundlagen des Geschäftsprozess-Managements erläutert. Dabei liegt der Schwerpunkt auf der Modellierung von Geschäftsprozessen. Es wird eine ausgewählte Sprache zur Geschäftsprozess-Modellierung vorgestellt. Auf der Grundlage dieser Modellierungssprache werden Geschäftsprozess-Modelle einfacher Komplexität unter Verwendung eines Werkzeuges entwickelt. Außerdem werden Geschäftsprozess-Modelle im Hinblick auf verschiedene Kriterien wie z. B. Prozesszeit, Prozessqualität oder Prozesskosten bewertet und verbessert.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung					

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

## **Titel des Moduls: Grundlagen der Automatisierungstechnik**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Grundlagen der Automatisierungstechnik (135110)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**  
Modulbeauftragte/r: Prof. Dr.-Ing. Bernd Kuhlenkötter  
Lehrende: Prof. Dr.-Ing. Bernd Kuhlenkötter

**Verwendung des Moduls**  
B.Sc. Angewandte Informatik

### **Vorkenntnisse**

### **Lernziele (learning outcomes)**

#### **Zielsetzung:**

- Die Studierenden sollen in der Lage sein, aktuelle Entwicklungen und Trends in der Automatisierungstechnik darzulegen sowie Entwicklungsprozesse für automatisierte technische Systeme erläutern und die entsprechenden Entwicklungsmethoden anwenden zu können.
- Sie sollen durch Absolvieren des Kurses in die Lage gebracht werden, das Funktionsprinzip und den Hardware-Aufbau einer SPS darzulegen und Automatisierungsaufgaben im Bereich der SPS- und NC-Programmierung mit methodischer Vorgehensweise zu bearbeiten.
- Zudem sollen sie die Kenntnisse erlangen, Robotersysteme für den Einsatz in unterschiedlichen Automatisierungsaufgaben kritisch zu bewerten, geeignete Systeme auszuwählen sowie Sicherheitsrisiken der Automatisierungstechnik zu beurteilen.

#### **Kenntnisse:**

- Die Studierenden kennen wesentliche Methoden und Verfahren der Ingenieurwissenschaften / des Maschinenbaus, verfügen über entsprechendes Fachvokabular und kennen Anwendungsbeispiele.
- Die Studierenden kennen vertiefte ingenieurwissenschaftliche Grundlagen im Bereich ihres Studienschwerpunkts.

#### **Fertigkeiten:**

- Die Studierenden verfügen über die Fähigkeit zu vernetztem und kritischem Denken.
- Die Studierenden praktizieren erste Ansätze wissenschaftlichen Lernens und Denkens.
- Die Studierenden können ingenieurtechnische Probleme modellieren und lösen.
- Die Studierenden können komplexe mathematische Problemstellungen in physikalischen Systemen (ggf. fachübergreifend) mit geeigneten Methoden lösen.
- Die Studierenden haben die Fähigkeit zu vernetztem und kritischem Denken ausgebaut und sind in der Lage etablierte Methoden und Verfahren auszuwählen und anzuwenden.

#### **Kompetenzen:**

- Die Studierenden verfügen über fachübergreifende Methodenkompetenz.
- Die Studierenden können Erkenntnisse / Fertigkeiten auf konkrete maschinenbauliche /ingenieurwissenschaftliche Problemstellungen übertragen.

### **Inhalt**

Die Vorlesung „Grundlagen der Automatisierungstechnik“ (GdA) stellt die Themen der industriellen Automatisierung mit dem Fokus auf der Industrierobotik dar. Grundlegende Anwendungsgebiete, wie der Einsatz von Industrierobotik in Lackierstraßen oder Schweißapplikationen, werden neben der historischen Entwicklung der Automatisierungstechnik aufgezeigt. Ein Schwerpunkt der Vorlesung sind die in Bezug auf Automatisierungsaufgaben häufig genutzten Steuerungen mittels SPS und NC/CNC. Neben der Erläuterung des Hardwareaufbaus und des Funktionsprinzips einer SPS werden in vorlesungsbegleitenden Übungen eigene SPS-Programme erstellt. Innerhalb der Steuerungen spielt die Signalverarbeitung von der Erfassung der Sensorsignale über die Verarbeitung und Ausgabe sowie die Art der Kommunikation der Daten untereinander eine wesentliche Rolle. Weiterhin werden innerhalb der Vorlesung Projektabläufe und Planungen von beispielhaften automatisierten Prozessen mit den Studierenden erarbeitet. Einen weiteren Schwerpunkt der Vorlesungs- und Übungseinheiten bilden die vermittelten Grundlagen zur industriellen Robotik. Dabei wird zunächst die Entwicklung der Industrierobotik dargelegt. Des Weiteren werden die wesentlichen Bestandteile eines Robotersystems gelehrt und verschiedene Industrierobotertypen und deren Einsatzgebiete in der Automatisierungstechnik vorgestellt. Die prinzipielle Funktionsweise von Robotersteuerungen wird in weiteren Vorlesungs- und Übungseinheiten vertieft. Die Lehrveranstaltung schließt mit einer Einführung in die Grundlagen der Kommunikationstechnik, Sensorik und Sicherheitstechnik im Themenfeld der Automatisierung ab. Die Inhalte der Vorlesung bereiten Studierende auf die Arbeit als Automatisierungsingenieur vor. Vorträge von Gastreferenten aus Industrie und Forschung zeigen praxisnahe Anwendungsbeispiele aus der Automatisierungstechnik auf und ergänzen somit die Lehrveranstaltung. Übungen dienen der weiteren Vertiefung des gelesenen Lehrstoffes.

### **Lehrformen**

Vorlesung mit Übung

### **Prüfungsformen**

Schriftliche Modulabschlussprüfung (180 Minuten)

### **Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung.

### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

**Titel des Moduls: Highlights of Theoretical Computer Science [B.Sc.]**  
**Highlights of Theoretical Computer Science [B.Sc.]**

<b>Modul-Nr./Code</b>	<b>Credits</b> 10 CP	<b>Workload</b> 300 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Highlights of Theoretical Computer Science (211057)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 210	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Successful completion of an introductory course on theoretical computer science (covering formal languages, basics of complexity theory including NP-completeness and reductions, basics of computability theory). Interest and motivation to learn about theoretical concepts.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Michael Walter Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Michael Walter Prof. Dr. Thomas Zeume Dr. Vladimir Lysikov					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit / Informationstechnik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> You will know some of the most important results and insights of modern theoretical computer science. You will learn approaches and techniques that go well beyond a first course. You will be able to recognize when these can be used and how to adapt them to new situations. You will be able to independently acquire new knowledge in this area.					
<b>Inhalt</b> The insights and techniques of modern theoretical computer science have been key for advances in all areas of computer science. In this course, we will discuss some highlights and the techniques that underpin them.  Possible topics that we might cover: <ul style="list-style-type: none"> <li>• Computational models (is there life beyond Turing machines?)</li> <li>• Kolmogorov complexity (what is the shortest program that produces some output?)</li> <li>• Communication complexity (how many bits must Alice and Bob exchange to jointly solve a problem?)</li> <li>• Fine-grained complexity (are some easy problems easier than others? and why?)</li> <li>• Fast multiplication of numbers and matrices (can you beat the high-school method?)</li> <li>• Randomness (does it really help to compute faster?)</li> <li>• Circuit lower bounds (why is it so hard to prove that problems are hard?)</li> <li>• Convex optimization (how to maximize profit if all you can ask are yes/no questions)</li> <li>• Hardness of approximation (how easy is it to find near-optimal solutions?)</li> <li>• Cryptography and computation</li> </ul>					

If you enjoyed your first course in theoretical computer science in the Bachelor's and would like to deepen your knowledge by getting an overview of the fascinating theory of computing, then this course will be exactly right for you.

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Final module examination. Format will depend on number of participants.

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

10/158: B.Sc. Informatik [PO 22]

10/170: B.Sc. Informatik [PO 20]

10/168: B.Sc. Angewandte Informatik [PO 22]

10/170: B.Sc. Angewandte Informatik [PO 20]

10/150: B.Sc. IT-Sicherheit / Informationstechnik [PO 22]

10/149: B.Sc. IT-Sicherheit / Informationstechnik [PO 20]

**Titel des Moduls: Inklusives IT-Design**  
**Inclusive IT design**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Inklusives IT-Design (260020)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Christian Meske  
 Lehrende: Prof. Dr. Christian Meske  
 Markus Jelonek, M.Sc.

**Verwendung des Moduls**

B.Sc. Angewandte Informatik

M.Sc. Angewandte Informatik [nur im SS 23 als Vertiefungsmodul, ansonsten als Freies Wahlmodul]

**Vorkenntnisse**

**Lernziele (learning outcomes)**

Die Veranstaltung richtet sich an Studierende mit Interesse an Themen der Inklusion und dem Nutzerzentrierten Design.

Die Studierenden

- ... kennen die Perspektiven von Endnutzer:innen mit geistiger und/oder körperlicher Behinderung hinsichtlich Interaktionsdesign, Usability und User Experience und können diese bei der Entwicklung eines Prototyps anwenden, begründen und beurteilen.
- ... leisten einen Dienst an der Gesellschaft, indem sie reale Probleme bzgl. der Inklusion von Menschen mit Behinderungen im Rahmen der digitalen Transformation aufgreifen, analysieren, beurteilen und gemeinsam prototypische Lösungen konstruieren (Service Learning).
- ... können partizipative und Design Science Methoden mit Aspekten der Inklusion und des Wohlbefindens in zukünftigen IT-Design-Prozessen zusammenführen und beurteilen.
- ... können theoretisch-konzeptionelle Design-Science-Vorgehensmodelle mit einer praxisorientierten Anwendung im Themenfeld des Positive Computing anwenden.
- ... kennen Anforderungen an inklusives IT-Design und können diese praktisch anwenden.
- ... können Fachwissen, welches in anderen Fächern erworben wurde (UX, Software-Ergonomie und Usability Engineering, Positive Computing, etc.), problemorientiert anwenden und beurteilen sowie neues Wissen selbstständig aneignen.
- ... können zielgerichtet und interdisziplinär zusammenarbeiten sowie Lösungen in einem festen Zeitraum eigenständig konstruieren, analysieren und präsentieren.

- ... können die Vorteile und Herausforderungen bei der Entwicklung von Anwendungen mit und für die spezifische Zielgruppe beschreiben.
- ... können Aspekte des agilen IT-Projektmanagements für Ihren Praxisteil anwenden.
- ... können Aspekte des Positive Computing in ihrer Prototypenwicklung anwenden.

### **Inhalt**

- Inclusive und accessible IT-Design
- Universal Design vs. Inclusive and Accessible Design
- Barrierefreiheit
- UX und Usability in inclusive und accessible IT-Design
- Design Science Research
- Partizipative Methoden und Co-Creation

### **Lehrformen**

- Vorlesung (Flipped Classroom, hybrid) mit praktischen Phasen in Kooperation mit Studierenden der Hochschule Ruhr-West (digital und analog)
- Praxisorientierte Blockveranstaltungen und Projektphasen mit Praxispartner vor Ort und digital

### **Prüfungsformen**

Projektarbeit

### **Voraussetzungen für die Vergabe von Credits**

- Regelmäßige Teilnahme an Pflichtterminen
- Bestandene Zwischen- und Abschlusspräsentation
- Bestandene Abgabe der Dokumentation und Präsentation

### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

6/168: B.Sc. Angewandte Informatik [PO22]

6/170: B.Sc. Angewandte Informatik [PO20]



**Titel des Moduls: Introduction to Computational Neuroscience**  
**Introduction to Computational Neuroscience**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Introduction to Computational Neuroscience (211046)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Sen Cheng Lehrende: Prof. Dr. Sen Cheng					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>  - essentiell: (Higher) Mathematics 1 und 2, Statistics, Programming - Neuroscience (not required, but nice to have)					
<b>Lernziele (learning outcomes)</b>					
<b>Inhalt</b>					
<b>Lehrformen</b>  Vorlesung mit Übung					
<b>Prüfungsformen</b>					
<b>Voraussetzungen für die Vergabe von Credits</b>					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/168: B.Sc. Angewandte Informatik [PO 22]  5/170: B.Sc. Angewandte Informatik [PO 20]					

**Titel des Moduls: Künstliche Neuronale Netze (kein Angebot im WS 24/25)**  
**Artificial Neural Networks (no offer in WS 24/25)**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Artificial Neural Networks (212006)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 150 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Sen Cheng Lehrende: Prof. Dr. Sen Cheng					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik  M.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> Grundkenntnisse in der Infinitesimalrechnung, linearen Algebra, Statistik und Informatik. Erfahrung mit einer höheren Programmiersprache.					
<b>Lernziele (learning outcomes)</b> Die mathematischen Grundlagen, Möglichkeiten und Beschränkungen überwachter Lernverfahren für Regression und Klassifikation mit künstlichen neuronalen Netzen (KNN), sowie für deren Anwendung erforderliche praktische Kenntnisse werden vermittelt.  Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verstehen Studierende die theoretisch-mathematischen Grundlagen von KNN im Kontext des überwachten Lernens.</li> <li>• können Studierende selbstständig zwischen verschiedenen KNN unterscheiden und in einer Anwendungssituation das geeignete Verfahren auswählen.</li> <li>• können Studierende grundlegende Verfahren selbstständig in einer höheren Programmiersprache implementieren, sowie ihre eigene Implementierung und Standard- Implementierung anderer auf Daten anwenden.</li> <li>• können Studierende Ergebnis der KNN selbstständig interpretieren, insbesondere erkennen, wann sie unrealistisch sind.</li> </ul>					
<b>Inhalt</b> Verfahren: Struktur von Optimierungsproblemen, Regression, logistische Regression, biologische neuronale Netze, Modellselektion, universelle Approximationstheorem, Perzeptron, mehr-schichtiges Perzeptron, Backpropagation, tiefe neuronale Netze, rekurrente neuronale Netze, Long-Short Term Memory, Hopfield Netze, Boltzmann-Machine Software: python, numpy, matplotlib, scikit-learn, tensorflow					
<b>Lehrformen</b> Vorlesung, Hausaufgaben, angeleitete Übungen am Computer					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

6/168: B.Sc. Angewandte Informatik [PO 22]

6/170: B.Sc. Angewandte Informatik [PO 20]

6/158: B.Sc. Informatik [PO 22]

6/170: B.Sc. Informatik [PO 20]

6/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

6/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]

<b>Titel des Moduls: Mathematics for Modeling and Data Analysis</b> Mathematics for Modeling and Data Analysis					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Mathematics for Modeling and Data Analysis (211047)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 30 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b> Grundkenntnisse in Infinitesimalrechnung und linearer Algebra.					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss dieses Kurses <ul style="list-style-type: none"> <li>• kennen die Teilnehmenden das in diesem Kurs behandelte Material, siehe Inhalt,</li> <li>• haben sie ein intuitives Verständnis der grundlegenden Konzepte und können damit arbeiten,</li> <li>• können sie sich über all dies auf Englisch verständigen.</li> </ul>					
<b>Inhalt</b> Dieser Kurs behandelt mathematische Methoden, die für die Modellierung und Datenanalyse relevant sind. Besonderer Wert wird auf ein intuitives Verständnis gelegt, das für eine kreative Beherrschung der Mathematik erforderlich ist. Die folgenden Themen werden behandelt: <ul style="list-style-type: none"> <li>• Funktionen und deren Visualisierung</li> <li>• Vektorräume</li> <li>• Matrizen als Transformationen</li> <li>• Systeme von linearen Differentialgleichungen</li> <li>• Qualitative Analyse von nichtlinearen Differentialgleichungen</li> <li>• Bayessche Theorie</li> <li>• Markov-Ketten</li> </ul>					
<b>Lehrformen</b> Dieser Kurs wird nach dem Konzept des "flipped/inverted classroom" durchgeführt. Zunächst arbeiten die Studierenden das Online-Material selbständig durch. In der Vorlesungszeit diskutieren wir dann den Stoff, finden Verbindungen zu anderen Themen, stellen Fragen und versuchen, sie zu beantworten. Im Tutorium wird das neu erworbene Wissen in analytischen Übungen angewendet und dadurch vertieft. Ich ermutige alle Studierenden, sowohl in der Selbstlernzeit als auch im Tutorium in Teams zu arbeiten.					
<b>Prüfungsformen</b> Schriftliche (digitale) Modulabschlussprüfung (90 Minuten).					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung.					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/97: M.Sc. Computer Science

## **Titel des Moduls: Menschenzentrierte Robotik**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Menschenzentrierte Robotik (136070)			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> 35 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		

### **Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr.-Ing. Bernd Kuhlenkötter

Lehrende: Prof. Dr. Annette Kluge, Prof. Dr.-Ing. Bernd Kuhlenkötter, Jun.-Prof. Dr. Laura Kunold

### **Verwendung des Moduls**

B.Sc. Angewandte Informatik

### **Vorkenntnisse**

Für den Kurs sollten die Studierenden Teamfähigkeit mitbringen und Interesse an interdisziplinären Themen haben, die über den ingenieurwissenschaftlichen Bereich hinausgehen (wie z.B. die psychologischen Implikationen der Robotik).

### **Lernziele (learning outcomes)**

Zielsetzung:

- Alle Studierenden haben Grundkenntnisse über die Entwicklungen, Anwendungsbereiche und die aktuellen Trends im Bereich der Mobilen und Servicerobotik.
- Die Studierenden der Ingenieurwissenschaften verstehen die Funktionsweise und den Aufbau von Robotersystemen und sind in der Lage diese zu programmieren
- Sie haben Grundkenntnisse über Forschungsmethoden der Mensch-Roboter-Interaktion, Mensch-Roboter-Kollaboration und sind in der Lage Gestaltungsempfehlungen auf Basis empirischer Befunde abzuleiten.
- Alle Studierenden sind in der Lage die multimediale Landschaft zur Kommunikation zwischen Roboter und Mensch differenziert zu betrachten.
- Sie haben die Fähigkeit kleinere Projekte selbständig innerhalb einer Gruppe zu planen und durchzuführen.

Kompetenzen:

- Die Studierenden verfügen über fachübergreifende Methodenkompetenz.
- Sie erlernen die Arbeit in interdisziplinären Teams.
- Sie können Erkenntnisse/Fertigkeiten auf konkrete maschinenbauliche/ingenieurwissenschaftliche Problemstellungen übertragen.
- Sie lernen Grundlagen anderer Disziplinen im interdisziplinären Austausch kennen.
- Sie verfügen über experimentelles Fachvokabular, kennen empirische Forschungsmethoden und die Grundlagen der Sozialpsychologie.
- Sie können disziplinübergreifende Inhalte aus nicht technischen Bereichen reflektieren und verantwortungsbewusst neue Ansätze in den Projektarbeiten entwickeln.
- Sie können durch die Projektarbeiten effektiv und effizient in Teams kommunizieren, diskutieren und ihre Arbeiten im Anschluss präsentieren

**Inhalt**

Die Studierenden werden in fachübergreifenden Gruppen an konkreten Problemstellungen im Bereich der menschenzentrierten Robotik arbeiten. Zur Gestaltung des sozio-technischen Systems aus Mensch(en) und Roboter(n), werden sowohl ingenieurwissenschaftliche als auch psychologische Fragen berücksichtigt. Vorbereitend hierfür wird aus technischer Sicht eine thematische Einführung in die Historie, Anwendungsfeldern und Funktionsweisen von Robotersystemen gegeben. Dabei wird vor allem auf die mobile Servicerobotik und die Mensch-Roboter-Kollaboration eingegangen. Zur menschengerechten Gestaltung der Interaktion mit der Roboterplattform, wird eine Einführung in psychologische Effekte der Mensch-Technik-Interaktion gegeben, sowie die soziale Robotik und ihre Anwendungsfelder vorgestellt.

Auf dieser Basis bearbeiten die Studierenden dann in interdisziplinären Gruppen individuelle Problemstellungen unter wissenschaftlichen Gesichtspunkten zur Weiterentwicklung einer mobilen Roboterplattform. Hierbei steht thematisch die erfolgreiche Kommunikation und Interaktion zwischen Mensch und Roboter (und der Eindruck des Roboters auf dem Menschen) im Vordergrund. Es besteht die Möglichkeit die ausgearbeiteten Lösungsansätze zu implementieren und somit eine Validierung des Konzeptes durchzuführen.

Die Studierenden werden bei der Projektplanung und dem Projektmanagement unterstützt, indem ihnen die Grundlagen des Projektablaufs für die jeweiligen Projektphasen vermittelt werden. Während der Projektlaufzeit wird durch Zwischengespräche die Projektentwicklung überprüft und reguliert. So werden die in der Lehrveranstaltung vorgestellten Methoden und das erlernte Wissen praktisch angewendet und das Arbeiten in interdisziplinären Projektgruppen eingeübt.

**Lehrformen**

Vorlesung (2 SWS)

**Prüfungsformen**

Hausarbeit (semesterbegleitend)

**Voraussetzungen für die Vergabe von Credits**

- Bestandene Modulabschlussprüfung: Dokumentation und Präsentation (Hausarbeit)
- Teilnahme an allen Zwischengesprächen

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

6/168: B.Sc. Angewandte Informatik [PO 22]

6/170: B.Sc. Angewandte Informatik [PO 20]

<b>Titel des Moduls: Mensch-Maschine-Interaktion (nächstmalig SS 25)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> nächstmalig im SS 25	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Mensch-Maschine-Interaktion (260083)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Christian Meske Lehrende: Prof. Dr. Christian Meske					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Programmierkenntnisse					
<b>Lernziele (learning outcomes)</b> Die Studierenden erlernen Gestaltungsprinzipien der Mensch-Maschine-Interaktion über die semesterbegleitende Anwendung des Design Science Research.					
<b>Inhalt</b> Design Science Research Grundlegende Eigenschaften von Menschen und Arbeitskontexten: Wahrnehmungs- und Kognitionspsychologie, Physiologie und Motorik Grundlagen des UI- und Interaktionsdesigns Nutzer:innen-Anforderungen erheben, verstehen und prototypisch umsetzen Durchführung von Evaluationen Spezielle Themen der MMI: Virtual Reality, Augmented Reality, Ubiquitous Computing					
<b>Lehrformen</b> Vorlesung inkl. Online-Inhalte, sowie Übung mit Projektarbeit. Die Veranstaltung wird hybrid durchgeführt: in Vorlesung und Übung wird es sowohl Präsenz- als auch Online-Termine per Zoom geben.					
<b>Prüfungsformen</b> Vorlesungsbegleitende Projektaufgabe: benotete Abschlusspräsentation, Abgabe Kurzbericht (muss bestanden sein)					
<b>Voraussetzungen für die Vergabe von Credits</b> &#8226; Regelmäßige Teilnahme an den Übungen &#8226; Zwischen- und Abschlusspräsentation zur Projektaufgabe					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/168: B.Sc. Angewandte Informatik [PO 22] 5/170: B.Sc. Angewandte Informatik [PO 20]					



<b>Titel des Moduls: Methoden der Bioinformatik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Methoden der Bioinformatik: Vorlesung (190502)  Methoden der Bioinformatik: Übung (190522)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Axel Mosig Lehrende: Prof. Dr. Axel Mosig					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Grundkenntnisse in Biologie					
<b>Lernziele (learning outcomes)</b> Die Studierenden erlernen interdisziplinäre Denkweisen und notwendige Grundkenntnisse, um aktuelle Forschungsthemen der Bioinformatik verfolgen zu können (Vorlesung). Anhand von Fragestellungen der Biologie werden Fähigkeiten des algorithmischen und statistischen Modellierens und Problemlösens erworben (Übungsaufgaben).					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Algorithmische und statistische Grundlagen der Bioinformatik (Reguläre Ausdrücke, Endliche Automaten, Turing Maschinen, Komplexität, Dynamische Programmierung, Maximum Likelihood, Hidden Markov Modelle, Poisson Prozesse)</li> <li>• Algorithmen zur Analyse von Sequenz und Struktur von Bio-Molekülen, Rekonstruktion evolutionärer Beziehungen zwischen Sequenzen und Strukturen</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/168: B.Sc. Angewandte Informatik [PO 22]  5/170: B.Sc. Angewandte Informatik [PO 20]					

## **Titel des Moduls: Methoden der Computerlinguistik**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Methoden der Computerlinguistik (050008)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Stefanie Dipper Lehrende: Prof. Dr. Stefanie Dipper					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Die Veranstaltung richtet sich an Studierende der Angewandten Informatik ohne Vorkenntnisse in der Linguistik. Der parallele Besuch der Veranstaltung "Einführung in die Linguistik" wird empfohlen.					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über grundlegendes Wissen über Computerlinguistik: <ul style="list-style-type: none"><li>• Sie kennen die wichtigsten, grundlegenden Analyseebenen in der Computerlinguistik und ihre spezifischen Eigenschaften.</li><li>• Sie kennen verschiedene, insbesondere statistische Verfahren zur automatischen Analyse von Sprachdaten sowie die dazugehörigen effizienten Algorithmen, können diese implementieren und auf eine eigene Fragestellung anwenden.</li><li>• Sie kennen die spezifischen Probleme bei der automatischen Verarbeitung sprachlicher Daten (wie z. B. sparse data, Ambiguitäten) und können diese anhand konkreter Beispiele erklären.</li><li>• Sie kennen eine Reihe von Evaluationsverfahren sowie ihre Vor- und Nachteile und können diese anwenden.</li><li>• Sie kennen relevante Ressourcen und wissen, in welcher Form sie eingesetzt werden.</li></ul>					
<b>Inhalt</b> Dieses Modul führt in die Kernmethoden und -modelle der computerlinguistischen Forschung ein. Neben theoretischen Grundlagen der Computerlinguistik (Chomsky-Hierarchie) werden entsprechende Modellierungen der verschiedenen Komplexitätsebenen eingeführt. Dabei werden vorrangig statistische Verfahren behandelt, bei denen das System Informationen aus Daten lernt. Neben klassischen probabilistischen Verfahren werden auch Methoden des Deep Learning behandelt, das in den letzten Jahren vermehrt Anwendung in der maschinellen Sprachverarbeitung findet. Darüber hinaus soll das Modul die Teilnehmer/innen befähigen, sich Fachpublikationen zu computerlinguistischen Arbeiten von der Fragestellung über die Umsetzung bis hin zu den Ergebnissen zu erschließen und für Präsentationen und schriftliche Arbeiten angemessen aufzubereiten.					
<b>Lehrformen</b> Seminar mit Anteilen von Inverted Classroom Settings mit digital verfügbaren, primär videobasierten Lehrmaterialien zum Selbststudium. Praktische Programmierübungen.					
<b>Prüfungsformen</b> Die benotete Modulabschlussprüfung wird in Form von schriftlichen Hausaufgaben (Programmieraufgaben) sowie einer Implementation eines computerlinguistischen Systems inklusive Dokumentation abgelegt. Gruppenarbeiten sind bei geeigneten Themen nach Absprache mit den Dozenten möglich.					

### **Voraussetzungen für die Vergabe von Credits**

Erfolgreicher Abschluss der Modulprüfung sowie bestandene Studienleistungen (Hausarbeiten).

In der Lehrveranstaltung steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht Anwesenheitspflicht und Sie sollten nach Möglichkeit nicht fehlen. Das bedeutet im Einzelnen:

Erlaubte Fehlzeiten im Modul Methoden der Computerlinguistik: max. 3 Sitzungen im Theorie- und Praxisteil.

Über diese Sitzungen hinaus dürfen Sie nur mit Entschuldigung (z. B. mit einem ärztlichen Attest) fehlen. In der ersten Sitzung besteht grundsätzlich Anwesenheitspflicht, weil dann die grundlegenden Informationen zum Ablauf des Kurses besprochen werden. Darüber hinaus können die Lehrenden einzelne Sitzungen, die z. B. für gemeinsames Arbeiten wesentlich sind, als verpflichtend erklären.

### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

**Titel des Moduls: Natural language processing with deep learning [B.Sc]**  
**Natural language processing with deep learning [B.Sc]**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 100 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Eigenes Laptop mit Python und einer geeigneten IDE für die Übungen mitbringen		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Ivan Habernal Lehrende: Prof. Dr. Ivan Habernal					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik					
<b>Vorkenntnisse</b> Praktisch: Kenntnisse in Python (Projekt-Strukturen, OOP, Unit-Tests), Kenntnisse in Numpy sind von Vorteil Theoretisch: Grundlagen der Analysis (Ableitungen) und der linearen Algebra (Vektoren), obwohl wir das wiederholen werden. Einige Grundkenntnisse der Linguistik (z. B. Syntax) auf Oberschulniveau sind ebenfalls von Vorteil.					
<b>Lernziele (learning outcomes)</b> Die Studierenden <ul style="list-style-type: none"> <li>• verstehen das Rückgrat des modernen NLP, wie z.B. Embeddings und Transformer-basierte Modelle</li> <li>• können die Fähigkeiten und Schwächen verschiedener Deep-Learning-Modelle im NLP kritisch beurteilen</li> <li>• verstehen verschiedene Tasks im NLP, deren Bewertung und Modellierungsannahmen</li> <li>• können verschiedene Modelle und Ansätze in Python implementieren und sammeln praktische Erfahrungen</li> </ul>					
<b>Inhalt</b> Typische Aufgaben und Datensätze der natürlichen Sprachverarbeitung (NLP) und ihre Bewertung. Warum ist NLP schwierig? Auffrischung der mathematischen Grundlagen, Calculus, gradientenbasierte Optimierung, Backpropagation für beliebige Funktionen. Log-lineare Modelle und Textklassifikation. Tiefe neuronale Netze. Sprachmodelle und Word embeddings. Lernen statischer Word embeddings. Rekurrente neuronale Netze. Encoder-Decoder, Text Generation, Attention und autoregressive Modelle. Transformers. Self-Attention und BERT. Reine Decoder-Modelle und GPT. LLMs: Prompting und In-context Lernen					
<b>Lehrformen</b> Wir werden Vorlesungen und Übungen haben. Die Vorlesungen werden ziemlich interaktiv sein, da versucht wird, die Studierenden in Fragen und kritisches Denken einzubeziehen. Die Folien für jede Vorlesung werden im Voraus hochgeladen, damit die Studierenden sie für ihre Notizen verwenden können. Jede Vorlesung enthält Links zu relevanten Forschungsarbeiten für diejenigen, die sich eingehender mit dem Thema befassen möchten. Die Vorlesungen sind überwiegend theoretisch, d. h. wir behandeln die wichtigsten Konzepte, Ideen und mathematischen Beschreibungen, aber nicht, wie man es in einem bestimmten Framework programmiert. In den praktischen Kursen werden wir genau das Gegenteil tun. Die Studierenden werden mit aktuellen Mainstream-Frameworks für Deep Learning in NLP, wie Pytorch oder Huggingface, experimentieren. Wir werden ein breites Spektrum an Komplexitäten abdecken, von der Programmierung eines einfachen neuronalen Netzes von Grund auf bis hin zur Verwendung eines vortrainierten Sprachmodells.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 90 Minuten					

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/168: B.Sc. Angewandte Informatik

5/158: B.Sc. Informatik

## **Titel des Moduls: Nebenläufige Programmierung (letztmalig SS 23)**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Nebenläufige Programmierung (211012)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr.-Ing. Doga Arinir (Lehrauftrag) Lehrende: Dr.-Ing. Doga Arinir					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  Master IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> Beherrschung einer Objektorientierten Programmiersprache (idealerweise Java)					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"><li>• haben die Studierenden grundlegende Fähigkeiten und Techniken erworben, um nebenläufige Programme sicher entwickeln zu können</li><li>• kennen die Studierenden softwaretechnische Entwurfsmuster, welche bekannte Probleme bei nebenläufigen Programmen, wie zum Beispiel die Verklemmung, vermeiden lassen</li><li>• können die Studierenden die Performanz von Programmen durch den Einsatz der nebenläufigen Programmierung verbessern</li><li>• sind die Studierenden in der Lage, bestehende Programme zu analysieren und mögliche Fehler zu erkennen</li><li>• können die Studierenden die Sprachmerkmale und Schnittstellen von JAVA für die nebenläufige Programmierung sicher anwenden</li></ul>					
<b>Inhalt</b> Moderne Hardware-Architekturen lassen sich nur durch den Einsatz nebenläufiger Programme richtig ausnutzen. Die nebenläufige Programmierung garantiert bei richtiger Anwendung eine optimale Auslastung der Hardware. Jedoch sind mit einem sorglosen Einsatz dieser Technik auch viele Risiken verbunden. Die Veranstaltung stellt Vorteile und auch Probleme nebenläufiger Programme dar und zeigt, wie sich die Performanz von Programmen verbessern lässt.  1. Nebenläufigkeit: Schnelleinstieg <ul style="list-style-type: none"><li>• Anwendungen vs. Prozesse</li><li>• Programme und ihre Ausführung</li><li>• Vorteile und Probleme von nebenläufigen Programmen (Verbesserung der Performanz, Synchronisation, Realisierung kritischer Abschnitte, Monitore, Lebendigkeit, Verklemmungen)</li></ul> 2. Threads in Java					

- 3. UML-Modellierung von Nebenläufigkeit
- 4. Neues zur Nebenläufigkeit in Java 5 und Java 6
- 5. Realisierung von Nebenläufigkeit 6. Fortschritte Java-Konzepte für Nebenläufigkeit
- 6. Fortschritte Java-Konzepte für Nebenläufigkeit

**Lehrformen**

Online Vorlesung mit begleitendem eLearning Kurs

**Prüfungsformen**

Schriftliche Modulabschlussprüfung über 90 Minuten

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]

5/91 Master IT-Sicherheit/ Informationstechnik

<b>Titel des Moduls: Numerische Mathematik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Numerischer Mathematik: Vorlesung (150106) Numerischer Mathematik: Übung (150107)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Markus Reineke Lehrende: Dr. Mario Lipinski					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte aus Höhere Mathematik 1 und 2					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende die wichtigsten Methoden der Ingenieurmathematik</li> <li>• können Studierende mathematische Problemstellungen in physikalischen Systemen erkennen und lösen</li> <li>• praktizieren Studierende erste Ansätze wissenschaftlichen Lernens und Denkens</li> <li>• verfügen Studierende über fachübergreifende Methodenkompetenz</li> </ul>					
<b>Inhalt</b> Grundlegende Methoden der numerischen Mathematik: Verfahren zur Lösung großer linearer Gleichungssysteme (Gauß-Verfahren, L-R-Zerlegung, Cholesky-Verfahren und Verwandte), Verfahren zur Lösung nichtlinearer Gleichungen und Gleichungssysteme, insbes. Newton-Verfahren mit Modifikationen, Verfahren zur Berechnung von Eigenwerten und Eigenvektoren, Lagrange-, Hermite- und Spline-Interpolation, Verfahren zur numerischen Integration, Numerische Lösung gewöhnlicher Differentialgleichungen, Anfangswertprobleme (Einschrittverfahren, insbes. Runge-Kutta-Verfahren, Ordnung und Konvergenz, Bedeutung der Stabilität und Anwendung auf steife Systeme, Schrittweitenkontrolle, Mehrschrittverfahren).					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (180 Min)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/168: B.Sc. Angewandte Informatik [PO 22] 5/170: B.Sc. Angewandte Informatik [PO 20]					





<b>Titel des Moduls: Programming for Modern Machine Learning</b> Programming for Modern Machine Learning					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> see examination regulations	<b>Turnus</b> Winter Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Programming for Modern Machine Learning (212040)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> English			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Nils Jansen Prof. Dr. Bilal Zafar Lehrende: Prof. Dr. Nils Jansen ( <a href="http://nilsjansen.org">http://nilsjansen.org</a> ) Prof. Dr. Bilal Zafar ( <a href="https://informatik.rub.de/zafar">https://informatik.rub.de/zafar</a> )					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik					
<b>Vorkenntnisse</b> Basic knowledge of statistics and programming.					
<b>Lernziele (learning outcomes)</b> At the end of this course, you will be able to: <ul style="list-style-type: none"> <li>Map a real world problem on a AI/ML paradigm like classification and reinforcement learning.</li> <li>Select the right model type for your task, e.g., XGBoost, MLP, CNN, Transformers.</li> <li>Select the right frameworks (scikit-learn, PyTorch, HuggingFace Transformers) and APIs (OpenAI API) for your tasks.</li> <li>Put together common components like data loaders and training loops to con-struct your projects.</li> <li>Use best practices like testing and linting to build maintainable code</li> </ul>					
<b>Inhalt</b> Past few years have seen confluence of two related trends: 1/ A rapid adoption of Artificial Intelligence (AI) and Machine Learning (ML) in a wide range of real-world applications across a variety of domains, e.g., healthcare, engineering. 2/ Development of specialized tooling and design patterns for AI/ML workloads. The goal of this course is to introduce the students to various AI/ML prediction paradigms, popular frameworks and design patterns. Specifically, we will build code bases involving (shallow) classification / regression models, CNNs and Transformers using frameworks like scikit-learn, PyTorch and Transformers. We will learn about using data loaders to manage large scale dataset and using GPUs to speed up deep learning workloads. We will also learn about best practices like testing and reproducibility.					
<b>Lehrformen</b> The course will be split into four blocks, each corresponding to a distinct paradigm of machine learning (traditional supervised learning, computer vision, language modelling and reinforcement learning). Each block will have corresponding in-person lectures and mini-projects.					
<b>Prüfungsformen</b> 50% e-Klausur (120 Minuten) + 50% Projektarbeit					
<b>Voraussetzungen für die Vergabe von Credits</b> Passing grade in the practical project and the final exam.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 6/168: B.Sc. Angewandte Informatik					



<b>Titel des Moduls: Studienprojekt</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> 5	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 210 h	<b>Gruppengröße</b> 5 Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: Dozierende im Studiengang Angewandte Informatik					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Gute Programmierkenntnisse (aus den Modulen Informatik 1 und 2, Software-Engineering) und ggf. weitere Kompetenzen je nach Wahl des Projektes					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls: <ul style="list-style-type: none"> <li>• können Studierende theoretischen Erkenntnisse zu Softwareentwicklungsprozessen, Projektabläufen und -Phasen an eigenen Projekten anwenden</li> <li>• haben Studierende Ihre Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung erweitert</li> <li>• haben Studierende Ihre Fähigkeit zur Teamarbeit verbessert</li> <li>• können Studierende Ihre Projektergebnisse sowohl mündlich als auch schriftlich darstellen und reflektieren</li> </ul>					
<b>Inhalt</b> Im Rahmen des Studienprojekts soll eine Aufgabe aus Bereichen der Angewandten Informatik in Teamarbeit unter Anleitung eines Betreuers gelöst werden. Die angebotenen Projekte decken dabei thematisch die gesamte Bandbreite der Vertiefungsmodule ab. Darüber hinaus werden auch interdisziplinäre Studienprojekte angeboten. In diesen arbeiten z.B. Angewandte Informatiker mit Studierenden der Sozialwissenschaften oder der Sportwissenschaft zusammen.					
<b>Lehrformen</b> Betreutes Projekt					
<b>Prüfungsformen</b> Projektarbeit mit Abschlussbericht					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Projektarbeit und fristgerechte Abgabe des Abschlussberichts					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 8/168: B.Sc. Angewandte Informatik					

**Titel des Moduls: System Performance Evaluation (keine Angebot im WS 24/25)**  
**System Performance Evaluation (no offer in WS 24/25)**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> System Performance Evaluation (212033)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf Lehrende: Prof. Dr.-Ing. Steffen Bondorf					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Empfohlen: Inhalte der Module Technische Informatik 1 und Computernetze					
<b>Lernziele (learning outcomes)</b> Die Studierenden sollen eine breite Kenntnis über folgende Themenfelder erlangen: <ul style="list-style-type: none"> <li>• Systemmodellierung und -leistungsanalyse</li> <li>• Entwurf von Experimenten</li> <li>• Messungen</li> <li>• Simulation</li> </ul>					
<b>Inhalt</b> Die Veranstaltung „System Performance Evaluation“ vermittelt grundlegende Techniken für Modellierung von Computersystemen und der Quantifizierung ihrer Leistung. Im Vordergrund stehen der Entwurf von Experimenten, Simulation und Warteschlangen-theorie.					
<b>Lehrformen</b> Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner können weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
<b>Prüfungsformen</b> 50% semesterbegleitendes Projekt + 50% Abschlussprüfung (geplant mündlich)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandenes Projekt und bestandene Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  5/158: B.Sc. Informatik [PO 22]  5/170: B.Sc. Informatik [PO 20]  5/168: B.Sc. Angewandte Informatik [PO 22]					



## **Titel des Moduls: Technische Informatik 3**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 Stunden	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Technische Informatik 3 - Hardwareprogrammierung (212003)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>  keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Tim Güneysu Lehrende: Prof. Dr.-Ing. Tim Güneysu					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte der Module Informatik 1 – Programmierung, Informatik 2 – Algorithmen und Daten-strukturen, Technische Informatik 1 – Rechnerarchitektur und Technische Informatik 2 – Digitaltechnik					
<b>Lernziele (learning outcomes)</b> Die Studierenden sollen Kenntnisse über technische Herausforderungen bei der anwendungsbezogenen Entwicklung von eingebetteten Systemen sowie des Internet of Things erlernen. Im Vordergrund der Veranstaltung stehen die maschinennahe Programmierung sowie die problemgerechte Integration von Aktorik und Sensorik. Nach dem erfolgreichen Abschluss des Moduls - kennen Studierende das Entwicklungs- und Programmiermodell sowie spezifische Eigenschaften von Mikrocontrollern als zentrale Grundlage eines eingebetteten Systems - haben Studierende die Fähigkeit zur maschinennahen Programmierung eines ausgewählten Mikrocontrollers zur Realisierung grundlegender Steuerprozesse sowie für die umgebende Peripherie 43 - erlernen Studierende Kommunikations- und Interaktionskonzepte (z.B. serielle Kommunikation via UART, SPI, PCIe etc.) mit externen Komponenten - können Studierende nebenläufige Prozesse strukturieren, in Systemen integrieren sowie damit verbundene potenzielle Probleme verstehen - sind Studierende in der Lage, komplexe anwendungsnahe Anforderungen mittels ausgewählter Hardwarekomponenten in ein eingebettetes System zu realisieren					
<b>Inhalt</b> Die Entwicklung von komplexen eingebetteten Systemen in Form einer zentralen Steuereinheit mit unterschiedlicher Sensorik und Aktorik spielen in vielen Anwendungen eine zentrale Rolle. In der Veranstaltung „Technische Informatik 3“ werden die vielfältigen Aufgabentypen und Realisierungsmöglichkeiten eines eingebetteten Systems sowie dessen anwendungsnaher Entwurf und Implementierung behandelt. Ein besonderes Gewicht in der Veranstaltung wird dabei auf ein anwendungsnahes Beispiel aus der Robotik gelegt, bei der viele der genannten Eigenschaften, Programme und Peripherie entwickelt und gesamtheitlich integriert werden müssen, um die Fähigkeiten eines einfachen Robotersystems auf vorhandener Hardware nachzubilden.					
<b>Lehrformen</b> Vorlesung (als Folien und Tafelvortrag) und Übungen, bei denen die vorgestellten Konzepte und Techniken praktisch an unterschiedlichen Hardware-Architekturen umgesetzt werden. Die Übungen beinhalten Elemente der Gruppen- und Projektarbeit					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 150 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

5/158: B.Sc. Informatik [PO 22]

5/165: B.Sc. Informatik [PO 20]

5/168: B.Sc. Angewandte Informatik [PO 22]

5/170: B.Sc. Angewandte Informatik [PO 20]



<b>Titel des Moduls: Vertiefungspraktikum (Angewandte Informatik)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 3 CP	<b>Workload</b> 90 h	<b>Semester</b> 4	<b>Turnus</b> Wintersemester oder Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>  In jedem Semester wird eine wechselnde Auswahl an Praktika bereitgestellt. Die zugeordneten Veranstaltungen können im Vorlesungsverzeichnis eingesehen werden.			<b>Kontaktzeit</b> Je nach Veranstaltungswahl	<b>Selbststudium</b> Abhängig von Praktikumswahl	<b>Gruppengröße</b> 15 Studierende
<b>Unterrichtssprache</b>  Abhängig von Praktikumswahl: Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan der Angewandten Informatik Lehrende: siehe jeweiligen Praktikumseintrag					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Gute Programmierkenntnisse, ggf. weitere Vorkenntnisse abhängig vom Praktikum					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>haben Studierende Ihre Fähigkeiten im Programmieren vertieft und erweitert</li> <li>je nach gewählten Programmierpraktikum können noch weitere Lernziele dazu kommen</li> </ul>					
<b>Inhalt</b> Es werden Praktika zu verschiedenen relevanten Themen angeboten, wie z.B. Grundlagen der Roboterprogrammierung, Erklärbare Künstliche Intelligenz - Programmierpraktikum oder Open Neural Data.  Weiterführende Informationen zu den jeweiligen Praktika finden Sie im Vorlesungsverzeichnis.					
<b>Lehrformen</b> Praktikum im Block oder als semesterbegleitende Veranstaltung.					
<b>Prüfungsformen</b> Praktikum					
<b>Voraussetzungen für die Vergabe von Credits</b> Erfolgreiche Teilnahme am Praktikum					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> unbenotet					

<b>Titel des Moduls: Vertiefungsseminar (Angewandte Informatik)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 3 CP	<b>Workload</b> 90 h	<b>Semester</b> 6	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> In jedem Semester wird eine wechselnde Auswahl an Seminaren bereitgestellt. Die zugeordneten Seminare können im Vorlesungsverzeichnis eingesehen werden.			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 60 h	<b>Gruppengröße</b> 15 Studierende
<b>Unterrichtssprache</b> Je nach Seminarwahl: Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: siehe jeweilige Seminare					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Die Vertiefungsseminare beziehen sich in der Regel auf Inhalte aus bestimmten Pflicht- oder Vertiefungsmodulen, die im Vorfeld absolviert worden sein sollten.					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verfügen Studierende über vertiefte wissenschaftliche Kenntnisse in dem ausgewählten Seminarthema</li> <li>• haben Studierende das Halten eines wissenschaftlichen Vortrags praktisch eingeübt und können Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln</li> <li>• können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen</li> </ul>					
<b>Inhalt</b> Es werden Bachelorseminare zu mehreren relevanten Themen angeboten, wie beispielsweise zu maschinellem Lernen, Algorithmen, theoretischer Informatik oder zu Ingenieurinformatik. Von den angebotenen Themen wählen die Studierenden abhängig von den eigenen Interessen und den individuellen Vertiefungswünschen ein Thema aus. Dieses sollen die Studierenden selbstständig bearbeiten. Dazu gehören die Literaturrecherche, die Einarbeitung in das Thema und schließlich die Präsentation. Nähere Informationen sind zu den jeweiligen Seminaren im Vorlesungsverzeichnis zu entnehmen.					
<b>Lehrformen</b> Seminar					
<b>Prüfungsformen</b> Seminarvortrag					
<b>Voraussetzungen für die Vergabe von Credits</b> Der Seminarvortrag muss mindestens mit der Note „ausreichend“ bewertet sein.  Um die Lernziele zu erreichen, besteht im Seminar Anwesenheitspflicht an mindestens 9 von 10 Terminen. Mehrfaches Fehlen muss durch ein ärztliches Attest entschuldigt werden. Die Anwesenheit beim ersten Termin ist obligatorisch, da zu diesem Termin die Themen verteilt werden. Das Seminar gilt als nicht bestanden, wenn an mehr als einem Termin unentschuldigt gefehlt wurde.					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**  
3/168: B.Sc. Angewandte Informatik [PO 22]

**Titel des Moduls: Virtuelle Produktmodellierung und -visualisierung**  
**Virtual Product-Modelling and Visualization**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>  Virtuelle Produktmodellierung und -visualisierung (135060)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Detlef Gerhard Lehrende: Prof. Dr.-Ing. Detlef Gerhard					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b>  Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verfügen Studierende über ein breites, integriertes Wissen über die Herausforderungen moderner Produktentstehungsprozesse und die resultierenden Anforderungen an Softwaresysteme zur Virtuellen Produktmodellierung und -visualisierung.</li> <li>• kennen und verstehen Studierende wesentliche Methoden und Verfahren der Virtuellen Produktmodellierung und -visualisierung einschließlich der angrenzenden Gebiete und der wissenschaftlichen informationstechnischen Grundlagen. Indem sie praktische Beispiele und Aufgaben mit entsprechender Anwendungssoftware bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit Softwaresystemen auf konkrete konstruktionstechnische Problemstellungen übertragen, um diese modellieren und lösen zu können.</li> <li>• haben Studierende ein umfassendes Verständnis vom Zusammenwirken der Softwaresysteme und Produktdatenmodelle innerhalb der verschiedenen Prozessketten in der Produktentstehung und können kritisch die Eignung von Methoden zur Virtuellen Produktmodellierung und -visualisierung zur Konzeption, Konstruktion, Optimierung, Darstellung, Fertigungsvorbereitung und Dokumentation von Produkten differenzieren und beurteilen.</li> <li>• können Studierende Aufgabenstellungen der Virtuellen Produktmodellierung und -visualisierung reflektieren und bewerten sowie selbstgesteuert verfolgen.</li> <li>• können Studierende kooperativ Aufgabenstellungen der Virtuellen Produktmodellierung und -visualisierung in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen sowie über Sachverhalte umfassend kommunizieren.</li> </ul>					
<b>Inhalt</b>  Das Modul vermittelt Methoden und Werkzeuge zur "Virtuellen Produktmodellierung und -visualisierung", insbesondere das dazu erforderliche Grundlagenwissen und die relevanten methodischen Aspekte der systematischen Produktentwicklung. Schwerpunkte bilden dabei die verschiedenen CAD-Modellierungsmethoden (z.B. 3D Flächen und Volumenmodellierung, parametrische Modellierung, Baugruppenmodellierung) entsprechend der Anforderungen aus der Konstruktionsaufgabe sowie die Kombination von Verfahren zur durchgängigen Abbildung von Prozessketten (z.B. für Digital Mockup (DMU), Virtuelle und					

Augmentierte Realität (VR/AR), Auslegungs- und Nachweisberechnungen, Analyse und Simulation, Additive Manufacturing, Produktion (CAM), Digital Factory, Styling, Elektro/Elektronik-CAD) im Produktlebenszyklus mit Aspekten der Integration von Modellen und Werkzeugen.

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (90 Min., Anteil der Modulnote 100 %)

Studienbegleitende Aufgaben: Hausarbeiten (Sofern die Hausarbeiten vor der Modulabschlussprüfung absolviert werden, sind optional Bonuspunkte für die Klausur möglich) (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

**Voraussetzungen für die Vergabe von Credits**

- Bestandene schriftliche Modulabschlussprüfung
- Bestandene studienbegleitende Aufgaben: Hausarbeiten (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

Sowohl die Klausur als auch die Hausarbeit ist über Flexnow anzumelden.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**

<b>Titel des Moduls: Wirtschaft und Digitalisierung</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Wirtschaft und Digitalisierung (076040 und 076041)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 90 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Roland Düsing Lehrende: Dr. Roland Düsing					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Die TeilnehmerInnen des Moduls sollen <ul style="list-style-type: none"> <li>• die Bedeutung des Wirtschaftens erklären können,</li> <li>• die Bestandteile, Funktionen und Eigenschaften eines Markts beschreiben können,</li> <li>• digitale Güter charakterisieren können,</li> <li>• die Besonderheiten der Wertschöpfung auf digitalen Märkten erläutern können,</li> <li>• ausgewählte Marktmodelle darstellen können</li> </ul>					
<b>Inhalt</b> Die Digitalisierung hat Auswirkungen auf zahlreiche Bereiche des wirtschaftlichen Handelns. Digitale Produkte und Dienstleistungen, wie z. B. ePaper oder Cloud-Services, weisen im Vergleich zu materiellen Gütern spezielle Eigenschaften auf, die großen Einfluss auf das Kunden- und Wettbewerber-Verhalten auf digitalen Märkten haben. Die Digitale Transformation in Unternehmen ermöglicht Veränderungen der Wertschöpfung, die zu digitalen Geschäftsmodellen, wie beispielsweise Digitale Plattformen, führen können.  In diesem Modul werden die Auswirkungen der Digitalisierung auf das wirtschaftliche Handeln erläutert. Dabei werden zunächst die Grundlagen des Wirtschaftens dargestellt. Im Anschluss daran werden die Konzepte Digitalisierung und Digitale Transformation beschrieben. Schließlich werden die Besonderheiten der Wertschöpfung auf digitalen Märkten veranschaulicht und ausgewählte Marktmodelle skizziert.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/168: B.Sc. Angewandte Informatik [PO 22]  5/170: B.Sc. Angewandte Informatik [PO 20]					



<b>Titel des Moduls: Freie Wahlmodule</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b>	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> abhängig von Veranstaltungswahl	<b>Selbststudium</b> Je nach Veranstaltungswahl	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Je nach Veranstaltungswahl			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studienfachberatung Angewandte Informatik Lehrende:					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik [PO 22]					
<b>Vorkenntnisse</b> abhängig von Veranstaltungswahl					
<b>Lernziele (learning outcomes)</b> Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den freien Wahlfächern					
<b>Inhalt</b> Studierende müssen Veranstaltungen im Gesamtumfang von 9 CP absolvieren.  Je nach Veranstaltungswahl werden unterschiedliche Inhalte vermittelt.  Studierende haben die Möglichkeit unter den Freien Wahlmodulen auch Fächer jenseits der Informatik zu absolvieren und so ihre Soft Skills zu erweitern. Z.B. wird die englische Fachsprache verbessert, in die Grundlagen der Rechtswissenschaften eingeführt oder Grundkenntnisse der Betriebswirtschaft vermittelt. Die Studierenden haben die Möglichkeit eine Auswahl entsprechend der eigenen Interessen zu treffen.  Bei der Auswahl geeigneter Lehrveranstaltungen kann die Liste unter <a href="https://informatik.rub.de/studium/pruefungsamt/wahlbereich/">https://informatik.rub.de/studium/pruefungsamt/wahlbereich/</a> hilfreichen Input geben.					
<b>Lehrformen</b>					
<b>Prüfungsformen</b> abhängig von Veranstaltungswahl					
<b>Voraussetzungen für die Vergabe von Credits</b> abhängig von Veranstaltungswahl					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 0/168 (unbenotet): B.Sc. Angewandte Informatik [PO 22]					



<b>Titel des Moduls: Abschlussarbeit (Bachelor AI)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 15 CP	<b>Workload</b> 450 h	<b>Semester</b> 6	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Bachelorarbeit (12 CP) Kolloquium (3 CP)			<b>Kontaktzeit</b> 15 h	<b>Selbststudium</b> 435 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: Lehrende im Studiengang Angewandte Informatik					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Abhängig von der Themenwahl.					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls <ul style="list-style-type: none"> <li>• können Studierende selbstständig und fristgerecht ein wissenschaftliches Thema bearbeiten von der Recherche bis zur Dokumentation der Resultate</li> <li>• können Studierende geeignete wissenschaftliche Verfahren und Methoden, die sie im Studium kennengelernt haben, auswählen und anwenden, um ein konkretes Problem zu lösen</li> <li>• können Studierende ihre Ergebnisse kritisch mit dem Stand der Forschung vergleichen und evaluieren</li> <li>• können Studierende ihre eigenen Ergebnisse angemessen in Wort und Schrift darstellen</li> </ul>					
<b>Inhalt</b> Die Bachelorarbeit ist eine schriftliche Prüfungsarbeit. Es wird die selbstständige Bearbeitung einer anspruchsvollen Aufgabe der Angewandten Informatik unter Anwendung der im Bachelorstudium erworbenen Kenntnisse und Methoden erwartet. Im Anschluss an die Bearbeitung der Bachelorarbeit (12 CP) werden die Ergebnisse in Form eines Kolloquiumsvortrags (3 CP) präsentiert.					
<b>Lehrformen</b> Abschlussarbeit					
<b>Prüfungsformen</b> Schriftliche Bachelorarbeit und Kolloquiumsvortrag					
<b>Voraussetzungen für die Vergabe von Credits</b> Sowohl die Bachelorarbeit als auch der Kolloquiumsvortrag müssen bestanden sein. Die jeweiligen Teilnoten gehen gewichtet nach den CP in die Modulnote ein.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 15/168: B.Sc. Angewandte Informatik					