

RUHR-UNIVERSITÄT BOCHUM

FACULTY OF COMPUTER SCIENCE

RUB

Modulhandbuch

für den Bachelorstudiengang

INFORMATIK

STAND 30.05.2023

<https://informatik-rub.de/bsc/>



Faculty of
Computer
Science

www.informatik.rub.de



Wo finde ich Hilfe während des Studium?

Wichtige Informationsquellen rund um das Studium der Informatik:

Fakultätswebseite:
<https://informatik.rub.de/>

Studiengangsw Webseite:
<https://informatik.rub.de/studium/studiengaenge/inf/>

Studienfachberatung der Fakultät für Informatik:
<https://informatik.rub.de/studium/studienberatung/>

Prüfungsamt der Fakultät für Informatik:
<https://informatik.rub.de/studium/pruefungsamt/>

Fachschaftsrat Informatik:
<https://www.ruhr-uni-bochum.de/fsr-informatik/>

Bei fachlichen Fragen besteht die Möglichkeit die Dozenten während Ihrer Sprechstunden (siehe individuelle Webseiten) zu kontaktieren.

Weitere wichtige Kontaktadressen auf dem Campus sind:

Zentrale Studienberatung:
<https://studium.ruhr-uni-bochum.de/de/zentrale-studienberatung>
Bietet Hilfe und Coaching bei individuellen Problemen (auch psychologische Betreuung).

Studienfinanzierungsberatung:
<https://studium.ruhr-uni-bochum.de/de/studienfinanzierung>

Stipendienberatung der RUB:
<https://studium.ruhr-uni-bochum.de/de/stipendienberatung>
Beratung zu Stipendien für Studieninteressierte und Studierende.

Beratungszentrum zur Inklusion Behinderter und chronisch Kranker:
<https://www.akafoe.de/inklusion/>

International Office:
<https://international.ruhr-uni-bochum.de/de/auslandsaufenthalte-im-studium>
Beratung zu Studienaufenthalten im Ausland

Wohnheimplätze:
<https://www.akafoe.de/wohnen/>

Studiengangsziele:

Der Bachelorstudiengang *Informatik* ist ein wissenschaftlicher Studiengang, der sich an die Studieninteressierte richtet, die mathematisch, naturwissenschaftlich und technisch interessiert sind, ihren Schwerpunkt auf die Informatik setzen und eine wissenschaftliche Laufbahn oder eine spätere Tätigkeit in der Industrie aufnehmen wollen. Es handelt sich um einen Vollzeit- und Präsenzstudiengang mit einer Regelstudienzeit von sechs Semestern. Die Absolvent*innen erlangen den Abschluss „*Bachelor of Science*“ (abgekürzt B.Sc.) und die Berufsbezeichnung Ingenieur*in.

Das Studium vermittelt das breite und in ausgewählten Teilgebieten vertiefte fachliche Wissen, um analytisch, kreativ und konstruktiv technische Systeme aus Soft- und Hardware zu konzipieren, zu entwickeln und zu warten. Auch die Rolle der Daten als wichtige Ressourcen in der heutigen Welt, inklusive deren Digitalisierung, Analyse durch leistungsstarke Technologien und Grundlage zum maschinellen Lernen und für intelligente Systeme sind Ausbildungsziele.

Die Informatik ist international und ihre „Amtssprache“ ist Englisch. Daher ist es ein Ziel des Studienganges, die Sprachkompetenzen der Studierenden durch Nutzung der englischen Sprache als zweite gleichberechtigte Sprache im Studium auszubauen. Während die Grundlagen in den ersten drei Semestern überwiegend auf Deutsch unterrichtet werden, ist die Unterrichtssprache in der zweiten Hälfte des Studiums vermehrt Englisch.

Modularisierungskonzept:

Das Studium ist modular aufgebaut. Ein Modul ist eine inhaltlich und zeitlich abgeschlossene Lehr- und Lerneinheit und erstreckt sich über ein bis maximal zwei Semester. Durch die Modularisierung ist das inhaltliche Angebot des Studienprogramms strukturiert.

Jedem Modul ist eine bestimmte Anzahl an Leistungspunkten zugeordnet, die den Arbeitsaufwand für das erfolgreiche Abschließen des Moduls widerspiegelt. Mit einem Leistungspunkt wird der Workload von ca. 30 Stunden bei einem durchschnittlichen Studierenden bezeichnet, wobei die Präsenzzeiten auch berücksichtigt werden. Die durch einen erfolgreichen Modulabschluss zu vergebenden Leistungspunkte (LP) bzw. Credit Points (CP) entsprechen im Umfang dem Europäischen Credit Transfer System (1 LP = 1 CP = 1 ECTS).

Der zu absolvierende Workload des gesamten Studiengangs beträgt 180 Leistungspunkte mit durchschnittlich 30 Leistungspunkten pro Semester.

entwickeln.

Prüfungsformen:

Prüfungsleistungen können in Form einer Klausur (auch in elektronischer Form), einer mündlichen Prüfung, eines Seminarbeitrags, eines Referates oder Präsentation, einer Hausarbeit, einer Projektarbeit, einer praktischen Prüfung oder eines Kolloquiumsvortrags erbracht werden. Auch Kombinationen von verschiedenen Prüfungsformen sind möglich.

Studienverlaufsplan:

Nr	Modul	Umfang (LP)	Empfohlenes Semester	Bewertung
Pflichtbereich				
1	Mathematik 1	9	1	benotet
2	Informatik 1	12	1	benotet
3	Technische Informatik 1	5	1	benotet
4	English for Computer Science 1	3+3	1+2	benotet
5	Mathematik 2	9	2	benotet
6	Informatik 2	8	2	benotet
7	Technische Informatik 2	5	2	benotet
8	Programmierung und Programmiersprachen	6	2	benotet
9	English for Computer Science 2	3+3	3+4	benotet
10	Mathematik 3	9	3	benotet
11	Informatik 3	8	3	benotet
12	Software Engineering	5	3	benotet
13	Technische Informatik 3	5	3	benotet
14	Einführung in die künstliche Intelligenz	5	4	benotet
15	Datenbanksysteme	7	4	benotet
16	Computernetze	5	4	benotet
17	Betriebssysteme	5	4	benotet
18	Software Engineering Praktikum	5	4	benotet
Wahlpflichtbereich				
19	Vertiefungspraktikum**	3	5	unbenotet
20	Vertiefungsseminar**	3	5	benotet
21	Vertiefungsmodule***	20	5-6	benotet
Wahlbereich				
22	Freie Wahlmodule****	a*	1-6	unbenotet
Praktische Ausbildung				
23	Praktische Ausbildung*****	b*	5-6	unbenotet
Abschlussarbeit				
24	Bachelorarbeit und Kolloquium	12 + 3	6	benotet

Nr	Modul	Umfang (LP)	Empfohlenes Semester	Bewertung
Vertiefungsmodule				
1	Algorithmenparadigmen	5	WS	benotet
2	Autonomous Vehicles and Artificial Intelligence	5	SS	benotet
3	Digitale Forensik	5	WS	benotet
4	Einführung in die Kryptographie 1	5	WS	benotet
5	Einführung in die Kryptographie 2	5	SS	benotet
6	Game Development	6	SS	benotet
7	Information Theory	5	SS	benotet
8	Logik in der Informatik	5	WS	benotet
9	Model Checking	5	WS	benotet
10	Nebenläufige Programmierung	5	SS	benotet
11	Network Planing	5	WS	benotet
12	Proofs are programs	5	SS	benotet
13	Quantum Information and Computation	5	WS	benotet
15	Statistisches Lernen und Data Mining	5	SS	benotet
16	Systemsicherheit	5	SS	benotet
17	Web-Engineering	5	SS	benotet
Seminare				
1	Seminar Distributed and Networked Systems	3	WS	unbenotet
2	Seminar Implementation Security	3	WS	benotet
3	Seminar Machine Learning Applications	3	WS	benotet
4	Seminar Modern Programming Languages	3	WS	benotet
5	Seminar Reinforcement Learning	3	WS	benotet
6	Seminar Ressourceneffiziente Systemsoftwarekonzepte	3	WS/SS	benotet
7	Seminar From Biological to Artificial Neural Networks	3	SS	benotet
8	Seminar Satisfiability	3	SS	benotet
9	Seminar zu Approximationalgorithmen	3	SS	benotet
11	Seminar zu Knowledge Graphs	3	SS	benotet
10	Seminar Quantum Algorithms	3	SS	benotet
Praktika				
1	Praktikum Systemsoftware-Technik	3	WS	Nicht benotet
2	Praktikum Implementing Post Quantum Standards and Challenges	3	WS	Nicht benotet

3	Android App Evolution	3	WS	Nicht benotet
4	Introduction to Python	3	SS	Nicht Benotet
5	<u>Praktische Kryptanalyse von symmetrischen Chiffren</u>	3	SS	Nicht benotet
6	Praktikum IDE Plugin Development	3	WS	Nicht benotet
Praktische Ausbildung				
23	Praktische Ausbildung*****	b*	5-6	unbenotet
Abschlussarbeit				
24	Bachelorarbeit und Kolloquium	12 + 3	6	benotet

MODULHANDBUCH

Übersicht der Module

Informatik - Bachelor (1-Fach, PO 2022)

Pflichtbereich

Mathematik 1
Informatik 1
Technische Informatik 1
English for Computer Science 1
Mathematik 2
Informatik 2
Programmierung und Programmiersprachen
Technische Informatik 2
English for Computer Science 2
Mathematik 3
Informatik 3
Technische Informatik 3
Software Engineering
Betriebssysteme
Computernetze
Einführung in die künstliche Intelligenz
Datenbanksysteme
Software Engineering Praktikum

Wahlpflichtbereich

Algorithmenparadigmen
Autonomous Vehicles and Artificial Intelligence
Digitale Forensik
Einführung in die Kryptographie 1
Einführung in die Kryptographie 2
Game Development
Information Theory
Logik in der Informatik
Model Checking
Nebenläufige Programmierung
Network Planing
Proofs are programs
Quantum Information and Computation

Statistisches Lernen und Data Mining

Systemsicherheit

Web-Engineering

Vertiefungspraktikum Informatik

Vertiefungsseminar Informatik

Wahlbereich

Freies Wahlmodul

Praktische Ausbildung

Praktische Ausbildung

Bachelorarbeit

Abschlussarbeit

Titel des Moduls: Mathematik 1 Mathematics 1					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Mathematik 1 - Grundlagen			Kontaktzeit 105 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Erfolgreiches Bestehen der Modulklausur.		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Gregor Leander Lehrende: Prof. Dr. Gregor Leander					
Verwendung des Moduls Bachelor Informatik Bachelor IT-Sicherheit					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls -kennen Studierende grundlegende Begriffe und Schreibweisen der Mathematik - können Studierende die erlernten Techniken selbstständig anwenden und mathematische Sachverhalte darstellen - kennen Studierende die Grundlagen abstrakter mathematischer Strukturen und verschiedene Beispiele für Gruppen, Ringe und Körper - verstehen die Studierenden den abstrakten Vektorraum-begriff über beliebigen Körpern, können mit linearer Unabhängigkeit, Dimensionen und mit linearen Abbildungen umgehen - sind Studierende in der Lage, lineare Gleichungssysteme explizit zu lösen sowie Eigenwerte und Eigenvektoren zu berechnen					
Inhalt Dieses Modul gibt eine allgemeine Einführung in mathematische Grundlagen und behandelt wichtige Gebiete der Linearen Algebra. Folgende Themengebiete werden behandelt: <ul style="list-style-type: none"> • Grundlagen der Mathematik • Grundlegende mathematische Begriffe • Schreibweisen • Aussagenlogik • Mengenlehre • Relationen Algebraische Grundlagen • ganze Zahlen • Restklassen • Gruppen-, Ringe- und Körper-Axiome Lineare Algebra • Vektorräume • Basen • Dimension • Skalarprodukte • lineare Abbildungen • lineare Gleichungssysteme • Basiswechsel • Determinanten • Eigenwerttheorie 					
Lehrformen Vorlesung und Übungen					
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)

9/165: B.Sc. Informatik [PO 22]

Titel des Moduls: Informatik 1					
Modul-Nr./Code	Credits 12 CP	Workload 360 h	Semester 1	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen 212004 Informatik 1 (Vorlesung + Übung) 212400 Python-Praktikum			Kontaktzeit 6 Semesterwochenstunden + 10 Tage (80h)	Selbststudium 190 h	Gruppengröße 400 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers					
Verwendung des Moduls B.Sc. Angewandte Informatik B.Sc. Informatik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen die Teilnehmer die wichtigsten Konzepte imperativer und objektorientierter Programmierung, • können die Teilnehmer eigene Programme entwerfen und implementieren, • können die Teilnehmer mit Grundbegriffen der Informatik wie etwa Korrektheit, Laufzeit, Boole'scher Algebra, Invarianten und abstrakten Datentypen arbeiten, • können die Teilnehmer die einfache Datenstrukturen (Arrays, Dictionaries) gezielt einsetzen und kennen Standardalgorithmen darauf, insbesondere zum Sortieren von Arrays. 					
Inhalt <ul style="list-style-type: none"> • Imperative Programmierung (Variablen, Kontrollstrukturen, Funktionen und Rekursion, Fehlerbehandlung, Ereignisbehandlung) • einfache Datenstrukturen (Array und Dictionary) • Objektorientierung (Klassen, Sichtbarkeit, Schnittstellen, Vererbung) • Einführung in eine Reihe von Informatik-Konzepten (Invarianten, Laufzeitanalyse, Sortieralgorithmen, Repräsentation von Daten im Rechner, Boole'sche Algebra) 					
Lehrformen Vorlesung mit Übung plus zweiwöchiges Blockpraktikum Die Vorlesung nutzt das Flipped-Classroom Lehrformat. Sämtliches Vorlesungsmaterial steht online zur Verfügung.					
Prüfungsformen Schriftliche Modulabschlussprüfung Erfolgreiche Teilnahme am Praktikum					
Voraussetzungen für die Vergabe von Credits Teilnahme am Python-Praktikum und bestandene schriftliche Modulabschlussprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 12/168 B.Sc. Angewandte Informatik 12/158 B.Sc. Informatik					

Titel des Moduls: Technische Informatik 1
Technical informatics 1

Modul-Nr./Code	Credits 5 CP	Workload 150 Stunden	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Rechnerarchitektur			Kontaktzeit 4 SWS	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Philipp Niemann
 Lehrende: Prof. Philipp Niemann

Verwendung des Moduls
 B.Sc. Angewandte Informatik
 B.Sc. Informatik
 B.Sc. IT-Sicherheit / Informationstechnik

- Lernziele (learning outcomes)**
 Nach dem erfolgreichen Abschluss des Moduls
- kennen die Studierenden Zusammenhänge und haben Detailkenntnisse von den Komponenten und der Funktionsweise moderner Computersysteme. Dies schließt neben dem Prozessor auch das Speichersystem und die Schnittstellen zu weiteren Systemkomponenten ein
 - sind die Studierenden auf der Basis dieser Kenntnisse in der Lage, Computersysteme und deren Komponenten bezüglich verschiedener Metriken, wie z.B. Energieverbrauch, Rechenleistung, Speicherperformance etc. auf deren Eignung für eine bestimmte Aufgabe zu bewerten
 - haben die Studierenden die grundsätzliche Arbeitsweise und den prinzipiellen Aufbau von Prozessoren auf der Ebene der Mikroarchitektur verstanden und sind in der Lage, den Einfluss von Architekturmerkmalen, wie z.B. Pipelining oder Out-of-Order-Execution, auf die Befehlsausführung zu analysieren

Inhalt
 Die Veranstaltung Rechnerarchitektur befasst sich mit dem Aufbau und der Funktion moderner Prozessoren und Computersysteme. Ausgehend von grundlegenden Computerstrukturen wie der Von-Neumann- und der Harvard-Architektur werden der Aufbau, die Klassifizierung und die technische Realisierung von Rechnersystemen dargestellt. Hierbei wird die Programmierung auf Assemblerebene sowie die Verarbeitung von Programmen durch einen Prozessor erläutert. Darauf aufbauend folgen Methoden zu Leistungsbewertung von Prozessoren auf der Basis von standardisierten Benchmarks und verschiedene Metriken, um die Ergebnisse einordnen zu können.

Der inhaltliche Schwerpunkt der Vorlesung stellt die tiefgehende Analyse der Mikroarchitekturebene eines Prozessors dar, wobei sowohl der Datenpfad als auch das Steuerwerk im Rahmen der Vorlesung schrittweise entwickelt und erläutert werden. Auf der Basis des in der Vorlesung vorgestellten Prozessors werden dann moderne Verfahren zur Leistungssteigerung und deren Einsatzgebiete vorgestellt. Neben dem eigentlichen Prozessor wird auch das Speichersystem moderner Computer und verschiedene Schnittstellen zu internen und externen Komponenten des Computersystems behandelt.

Alle Themen werden mit aktuellen Beispielen aus verschiedenen Bereichen der Technik erläutert, sodass neben dem im Detail vorgestellten Beispielprozessor mit MIPS Architektur auch moderne Hochleistungsprozessoren mit x86-64 ISA, Prozessoren für eingebettete Systeme auf Basis der ARM-Architektur, extrem energiesparende Prozessoren auf Basis des MSP430, wie sie beispielsweise in IoT-Geräten zum Einsatz kommen, und anwendungsspezifische Spezialprozessoren auf Basis der Tensilica Xtensa Plattform vorgestellt werden.

Lehrformen
 Vorlesung (als Folien und Tafelvortrag) und Übungen, bei denen die vorgestellten Konzepte und Techniken praktisch umgesetzt werden, teilweise mit Rechnerübungen.

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)

5/158: 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]

Titel des Moduls: English for Computer Science 1					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester 1	Turnus Jedes Winter und Sommersemester	Dauer 2 Semester
Lehrveranstaltungen a) English for Computer Science I: Reading Skills Sprachkurs mit integrierten Übungen (2 SWS, 3 CP, im Wintersemester) b) English for Computer Science I: Listening Skills Sprachkurs mit integrierten Übungen (2 SWS, 3 CP, im Sommersemester)			Kontaktzeit 60 h	Selbststudium 120	Gruppengröße 25 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Einstufungstest des Zentrums für Fremdsprachenausbildung. Die Termine sind auf der folgenden Internet-Seite zu finden: http://www.zfa.ruhr-uni-bochum.de/lehre/einstufung/index.html.de Achtung: Diese Lehrveranstaltungen werden auf zwei verschiedenen Kompetenzniveaus angeboten: 1. Intermediate (B1/B2-B2) 2. Advanced (B2/C1-C2) Die Studierenden müssen sich für das ihrem Einstufungsergebnis entsprechende Kompetenzniveau anmelden. Bei einem Einstufungsergebnis unterhalb von B1/B2 muss zunächst ein Kurs der Kompetenzstufe B1 belegt werden, der kein Bestandteil von diesem Modul ist. Die Module a) und b) können unabhängig voneinander besucht werden. Es empfiehlt sich jedoch, mit dem Besuch von English for Computer Science 1 anzufangen.		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Frau Melissa Oldfield-Mariano Lehrende: Dozent*innen des Zentrums für Fremdsprachenausbildung (ZFA) der Ruhr-Universität Bochum, Melissa Oldfield-Mariano					
Verwendung des Moduls Dieses Modul wurde exklusiv für diesen Bachelor-Studiengang konzipiert. Daher ist eine Öffnung dieser Veranstaltung für Studierende anderer Studiengänge nicht vorgesehen.					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls haben die Studierenden das volle Kompetenzniveau B2 in der <i>Fachsprache</i> erreicht, d.h. in diesem Zusammenhang: - Die Studierenden verstehen englischsprachige Fachtexte in den von ihnen bereits erlernten Wissenschaftsbereichen - Die Studierenden verstehen englischsprachige Vorlesungen, Vorträge und Diskussionen soweit, dass sie ihnen problemlos folgen können - Die Studierenden beherrschen das wichtigste Fachvokabular (passiv und aktiv) und können dieses konform ihrer					

Fachkenntnisse innerhalb verschiedener Aufgaben/ Anforderungen im Studium einsetzen und anwenden

- Die Studierenden sind auf das Rezipieren von und Interagieren in fachlichen Veranstaltungen auf Englisch grundlegend vorbereitet
- Die Studierenden können anlassbezogen und zielführend in englischer Sprache im eigenen Fachkontext kommunizieren und agieren
- Demnach sind die Studierenden beispielsweise in der Lage, sich zu fachlichen Themen auszutauschen und an Diskussionen teilzunehmen
- Sie können Funktion und Form verschiedener Textsorten erarbeiten und dieses Wissen in selbstproduzierten Texten kompetent anwenden
- Sie weisen Kompetenzen im Bereich der sprachlichen Mediation auf und können sowohl Global- als auch Detailinformationen aus Hör- und Lesetexten anderen klar, präzise und prägnant vermitteln

Inhalt

In den Lehrveranstaltungen ‚**Reading Skills**‘ und auch ‚**Listening Skills**‘ erfolgt eine Einführung in die Wissenschaftssprache der Informatik. Anhand von zunehmend komplexer werdenden fachlichen Inhalten werden alle sprachlichen Teilkompetenzen auf dem Niveau B2 handlungsorientiert trainiert.

In der Lehrveranstaltung ‚**Reading Skills**‘ bauen die Studierenden schwerpunktmäßig ihre Kompetenzen im Bereich Leseverstehen und in der schriftlichen Produktion aus. Mit besonderem Fokus auf die Fachsprache lesen die Studierenden authentische Texte (Lehrwerkausschnitte, Artikel, Berichte etc.) und verfassen eigene studienbezogene Texte (z.B. Beschreibungen, Kommentare, Erklärungen, Anleitungen). In diesem Zusammenhang erfolgt ebenso eine Analyse von Funktionen von Texten und Formaten, Textbausteine werden verinnerlicht und zentrale Formulierungen automatisiert. Die Studierenden lernen auch, gelesene Fachtexte adäquat und korrekt schriftlich oder mündlich wiederzugeben (Mediation).

In der Lehrveranstaltung ‚**Listening Skills**‘ bauen die Studierenden schwerpunktmäßig ihre Kompetenzen im Bereich Hör-Sehverstehen und in der mündlichen Produktion aus. Anhand von authentischen Hör(Seh)texten (Vorträgen, Vorlesungen, Diskussionen u.a.) analysieren sie Unterschiede zwischen verschiedenen Hörtexten und trainieren Strategien zum Global- und Detailverstehen. Die Studierenden lernen, sowohl unkomplizierte wissenschaftliche Texte (studienniveaunkonform) zu verstehen und diese wiederzugeben (Mediation) als auch das Argumentieren zu technischen bzw. fachlichen Fragestellungen. Ebenso spielt die Beschreibung mathematischer Formeln und Prozesse eine Rolle.

Die beiden Lehrveranstaltungen haben verschiedene Kompetenzschwerpunkte, die sich gegenseitig ergänzen. Mit steigendem Studien- und Wissensniveau verstärkt sich ebenso die Progression in der englischen Sprache. Beide führen zu einem steten Auf- und Ausbau des fachlichen Vokabulars und haben zum Ziel, den Studierenden mehr Zutrauen beim Umgang mit wissenschaftlichen Texten und der eigenen sprachlichen mündlichen und schriftlichen Produktion zu geben.

Vorbereitende (im Sinne vom Flipped Classroom), vertiefende bzw. weiterführende Aufgaben in Moodle sind integrativer Bestandteil der Lehrveranstaltungen.

Lehrformen

Gruppenarbeiten, Sprachmittlung, Flipped Classroom, Blended Learning

Prüfungsformen

Semesterbegleitende Leistungsüberprüfung:

Zu a) English for Computer Science I: Reading Skills

eine Sammlung von Kursleistungen v.a. in den Teilkompetenzen Leseverstehen und Schreiben (z.B. Leseverstehentests, Erarbeitung schriftlicher Texte, Sprachmittlungsaufgaben (reading-into-writing))

Zu b) English for Computer Science I: Listening Skills

eine Sammlung von Kursleistungen v.a. in den Teilkompetenzen Hörverstehen und Sprechen (z.B. Hörverstehentests, Aufnahmen eigener Sprachproduktion, Gruppenprojekte)

Die Modulnote setzt sich aus dem Durchschnitt der Abschlussnoten der Moduleteile a) und b) im Verhältnis von 50% : 50% zusammen. Beide Moduleteile müssen mit mindestens einer 4,0 bestanden sein. Für die Berechnung der Abschlussnoten und der Modulnote wird die Skala des ZFA verwendet.

Ein nicht bestandener Moduleil muss wiederholt werden, d.h. die Lehrveranstaltung muss erneut belegt und erfolgreich abgeschlossen werden.

Voraussetzungen für die Vergabe von Credits

Zu a) und b)

- Anwesenheit von 75 Prozent
- erfolgreicher Abschluss aller semesterbegleitenden Leistungen

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]

Titel des Moduls: Mathematik 2 Mathematics 2					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Mathematik 2 - Algorithmische Mathematik			Kontaktzeit 105 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Christian Stump Lehrende: Prof. Dr. Christian Stump					
Verwendung des Moduls Bachelor Informatik Bachelor IT-Sicherheit					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende grundlegende Begriffe, Beweismethoden und Algorithmen aus der elementaren Zahlentheorie • können Studierende die Beweistechniken selbstständig anwenden und mathematische Sachverhalte darstellen • kennen Studierende erste Sätze und Methoden aus der Kombinatorik und insbesondere aus der Graphentheorie und verstehen deren strukturelle Eigenschaften • kennen Studierende erste fundamentale Algorithmen aus der Zahlentheorie und der Kombinatorik, können diese formalisieren, selbstständig implementieren sowie deren Laufzeiten analysieren 					
Inhalt Diese Lehrveranstaltung behandelt die folgenden Themen: - Euklidischer Algorithmus, Gruppen-, Ring-, Körperaxiome, Symmetriegruppen, Polynomarithmetik, formale Potenzreihen, modulare Arithmetik, Lemma von Bezout, Kleiner Satz von Fermat, diskreter Logarithmus, RSA-Verschlüsselungsverfahren, Primzahltests, Chinesischer Restesatz, p-adische Brüche, Newton-Verfahren, Asymptotische Notation durch Landausymbole, Binomialkoeffizienten, Rekursionsgleichungen, Erzeugendefunktionen, Prinzip der Inklusion-Exklusion, Vier-FarbenProblem, Dijkstra-Algorithmus, Satz von Cayley, Hamiltonkreise, Google PageRank Algorithmus, Satz von Perron-Frobenius. Konkrete Algorithmen werden in Computeralgebra-Systemen implementiert.					
Lehrformen Vorlesung mit Übungen					
Prüfungsformen Schriftliche Modulabschlussprüfung über 180 Minuten					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den praktischen Übungen am Rechner					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 9/165: B.Sc. Informatik [PO 22] 9/158: B.Sc. Informatik [PO 20]					

Titel des Moduls: Informatik 2 Informatics 2					
Modul-Nr./Code	Credits 8 CP	Workload 240 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen 211002: Informatik 2 - Algorithmen und Datenstrukturen			Kontaktzeit 90 h	Selbststudium 150 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Maïke Buchin Lehrende: Prof. Maïke Buchin					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik B.Sc. IT-Sicherheit / Informationstechnik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls: <ul style="list-style-type: none"> • können Studierende Algorithmen formal beschreiben und deren Korrektheit beweisen • können Studierende die Laufzeit und den Speicherbedarf von Algorithmen und Datenstrukturen analysieren und bewerten • kennen Studierende grundlegende Datenstrukturen • kennen Studierende grundlegende Schemata zum Entwurf von Algorithmen sind Studierende in der Lage, Algorithmen und Datenstrukturen für spezifische Probleme zu entwickeln • haben die Studierenden die Grundlagen der Programmiersprache Python kennengelernt 					
Inhalt 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).					
Lehrformen Hörsaalvorlesung mit Medienunterstützung und theoretische sowie praktische Übungen am Rechner.					
Prüfungsformen Schriftliche Modulabschlussprüfung über 150 Minuten					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 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]

Titel des Moduls: Programmierung und Programmiersprachen					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen 211053: Vorlesung + Übung			Kontaktzeit 60 h	Selbststudium 120 h	Gruppengröße 300 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan der Fakultät für Informatik Lehrende: Prof. Dr. Markus König					
Verwendung des Moduls B. Sc. Informatik B.Sc. Angewandte Informatik					
Lernziele (learning outcomes) Globales Ziel dieser Veranstaltung ist es, einen systematischen Überblick über Prinzipien, Methoden, Konzepte und Notationen der Objektorientierte Modellierung zu geben. Dieses Wissen – verbunden mit den praktischen Übungen am Computersystem – soll den Studierenden befähigen, nach internationalen Standards konzipierte, effiziente Programme problemgerecht zu entwickeln, zu analysieren und zu überprüfen. Dazu gehört das Beschreiben von Sachverhalten mittels der UML (Unified Modeling Language) und das Arbeiten mit der Programmiersprache Java. Besonders mit fortschreitender Digitalisierung ist es zentral, Programmstrukturen entwickeln zu können, welche einfach zu warten sind und es ermöglichen, umfangreiche Teams an einem Softwareprojekt zu beschäftigen. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • Können Studierende in der Programmiersprache Java programmieren. • Können Studierende Objektorientierte Methoden auf Programmiersprachen abstrahieren. • Können Studierende geläufige Objektorientierte Muster erkennen, anwenden und programmieren. • Sind Studierende in der Lage, umfassende Programmstrukturen eigenständig zu entwerfen, sodass diese übersichtlich und wartbar sind. • Können Studierende grafische Benutzeroberflächen in Java entwerfen. 					
Inhalt <ul style="list-style-type: none"> • Basiskonzepte der Objektorientierung • Klassen und Objekte • Generalisierung • Vererbung und Schnittstellen • Assoziationen • Generische Datentypen und Container • Unified Modeling Language • GUI-Programmierung • Ereignisverarbeitung • Model-View-Controller-Prinzip • Dialog- und E/A-Gestaltung 					
Lehrformen Vorlesungen, digitale und vor Ort betreute Übungen					
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)

6/158: B.Sc. Informatik

6/168: B.Sc. Angewandte Informatik

Titel des Moduls: Technische Informatik 2					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Technische Informatik 2 - Digitaltechnik für ITS und Informatik			Kontaktzeit 60 h	Selbststudium 90	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Amir Moradi Lehrende: Prof. Dr. Amir Moradi					
Verwendung des Moduls <ul style="list-style-type: none"> • Bachelor Informatik • IT-Sicherheit • Angewandte Informatik 					
Lernziele (learning outcomes) 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.					
Inhalt 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.					
Lehrformen Die Vorlesung wird als seminaristischer Unterricht abgehalten, die Übungen entweder am Rechner oder mit Stift und Papier.					
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) 5/158: B.Sc. Informatik [PO 22] 5/165: B.Sc. Informatik [PO 20]					

Titel des Moduls: English for Computer Science 2

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester 3	Turnus	Dauer 2 Semester
Lehrveranstaltungen a) English for Computer Science II: Presenting in English Sprachkurs mit integrierten Übungen (2 SWS, Wintersemester) b) English for Computer Science II: Writing in English Sprachkurs mit integrierten Übungen (2 SWS, Sommersemester)			Kontaktzeit 60 h	Selbststudium	Gruppengröße 25 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Bestandenes Modul <i>English for Computer Science I</i> Achtung: Diese Lehrveranstaltungen werden auf zwei verschiedenen Kompetenzniveaus angeboten: 1. Intermediate (B1/B2-B2) 2. Advanced (B2/C1-C2) Die Zuteilung zu den zwei Kompetenzniveaus im 2. Modul sollte mit der Zuteilung im 1. Modul übereinstimmen. Der Aufstieg in die höhere Kompetenzgruppe wäre nach einer sehr guten Leistung (Note) im ersten Modul möglich. Die Moduleile a) und b) können unabhängig voneinander besucht werden. Es empfiehlt sich jedoch, mit dem Besuch von <i>Presenting in English</i> anzufangen.		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Frau Melissa Oldfield-Mariano Lehrende: Dozent*innen des Zentrums für Fremdsprachenausbildung (ZFA) der Ruhr-Universität Bochum, Melissa Oldfield-Mariano					
Verwendung des Moduls Dieses Modul wurde exklusiv für diesen Bachelor-Studiengang konzipiert. Daher ist eine Öffnung dieser Veranstaltung für Studierende anderer Studiengänge nicht vorgesehen.					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> - können die Studierenden Fachbegriffe gezielt und effektiv in Präsentationen und schriftlichen Texten anwenden - sind die Studierenden in der Lage, ihr Wissen über ein Fachthema bzw. eigene Erkenntnisse sicher zu vermitteln und spontan dazu Fragen zu beantworten - sind die Studierenden in der Lage, in einer fachlichen Diskussion persönliche Standpunkte und Meinungen zu äußern und zu erfragen, Stellung zu beziehen, Hypothesen aufzustellen, Argumente und Gegenargumente zu formulieren sowie Vor- und Nachteile aufzuzeigen 					
Zu a) English for Computer Science II: Presenting in English Darüber hinaus beherrschen die Studierenden alle erforderlichen Kompetenzen, um eine (unkomplizierte) fachliche Präsentation im Englischen halten zu können, d.h. in diesem Zusammenhang: <ul style="list-style-type: none"> - Sie können aus verschiedenen Quellen Informationen für ihre Präsentation zusammentragen - Sie können diese Informationen fachkonform und den Konventionen der englischen Sprache angemessen 					

aufbereiten

- Sie sind in der Lage, die Vortragssprache (sprachliche Mittel) korrekt einzusetzen und die Präsentation publikumsadäquat vorzutragen
- Sie können auf Fragen zur Präsentation spontan eingehen
- Sie können mit fachrelevanten Vorträgen sicher umgehen

Zu b) English for Computer Science II: Writing in English

Die Studierenden beherrschen außerdem alle erforderlichen Kompetenzen, um (unkomplizierte) Fachtexte im Englischen schreiben zu können, d.h. in diesem Zusammenhang:

- Sie können aus verschiedenen Quellen Informationen für ihren Text zusammentragen
- Sie können diese Informationen fachkonform und den Konventionen der englischen Sprache angemessen aufbereiten
- Sie können ihre eigenen Texte überprüfen und über Verfasstes reflektieren.
- Sie können anderen Studierenden Feedback geben (mündlich und schriftlich) und fremde Texte evaluieren (Peer-Review)

Sie können mit fachrelevanten Textsorten sicher umgehen

Inhalt

Zu a) Presenting in English

Der Fokus dieser Veranstaltung liegt auf der mündlichen Kompetenz, hier insbesondere im Schwerpunktbereich des Präsentierens. Im Vordergrund steht eine Erweiterung des Wortschatzes und eine Verbesserung der fachlichen Ausdrucksweise im mündlichen Kontext. Authentische Vorträge und Vorlesungen aus dem Fach Informatik liefern sprachlichen Input und dienen als Grundlage für Analysen und als Beispiele für die Erstellung der eigenen Präsentation.

Innerhalb einer Projektarbeit erhalten die Studierenden die Gelegenheit, die fach-konformen lexikalischen und strukturellen Aspekte kennenzulernen und anzuwenden. Dabei wird das Erstellen und Durchführen von Präsentationen für unterschiedliche Adressaten eine Rolle spielen. Durch die regelmäßige Praxis und die intensive Arbeit mit Präsentationen gewinnen die Kursteilnehmenden eine größere Vertrautheit und Sicherheit im Umgang mit den fachrelevanten Vorträgen, entwickeln Präsentationsstrategien und erwerben Werkzeuge/Kompetenzen für eine effektive mündliche Kommunikation in der englischen Sprache in akademischen und beruflichen Kontexten.

Zu b) Writing in English

Der Fokus dieser Veranstaltung liegt auf der schriftlichen Kompetenz sowie einer Erweiterung des Wortschatzes und einer Verbesserung der Ausdrucksweise im fachlichen Kontext. Authentische Texte aus dem Fach Informatik liefern sprachlichen Input und dienen als Grundlage für Textanalysen und als Beispiele für die eigene schriftliche Produktion. Dadurch erhalten die Kursteilnehmenden die Gelegenheit, die fachkonformen lexikalischen und strukturellen Aspekte kennenzulernen und anzuwenden. Auch das Schreiben in verschiedenen Formaten und für unterschiedliche Adressaten wird eine Rolle spielen.

Zu den Aufgaben wird das wöchentliche Verfassen von Texten von einem vorgegebenen Umfang gehören, zu denen die Studierenden Feedback erhalten und die sie überarbeiten werden. Durch die regelmäßige Praxis und Anwendung der geschriebenen Sprache gewinnen die Kursteilnehmenden eine größere Vertrautheit und Sicherheit im Umgang mit den fachrelevanten Textsorten, entwickeln Schreibstrategien und erwerben Werkzeuge/Kompetenzen für einen effektiven schriftlichen Umgang mit der englischen Sprache in akademischen und beruflichen Kontexten.

Neben einer Steigerung der individuellen schriftlichen Fähigkeiten, arbeiten die Studierenden an Techniken des gegenseitigen Feedbacks (Peer-Review), erhöhen durch gezielte Übungen ihre schriftliche Ausdrucksfähigkeit im Englischen und werden an studiennahe Aufgabenformate herangeführt. In einem Portfolio werden die unterschiedlichen Schreibprodukte gesammelt, die den Lernprozess dokumentieren und dabei individuelle

Schwerpunkte sowie den Lernerfolg widerspiegeln.

Vorbereitende (im Sinne vom Flipped Classroom), vertiefende bzw. weiterführende Aufgaben in Moodle sind integrierender Bestandteil beider Lehrveranstaltungen.

Lehrformen

Gruppenarbeiten, Projektarbeit, Sprachmittlung, Flipped Classroom, Blended Learning

Prüfungsformen

Semesterbegleitende Leistungsüberprüfung:

Zu a) Presenting in English: wöchentliche Aufgaben zur Vorbereitung und abschließendes Abhalten einer Präsentation im vorgegebenen Format

Zu b) Writing in English: eine Sammlung von schriftlichen Aufgaben (Texte) unterschiedlicher Formate

Die Modulnote setzt sich aus dem Durchschnitt der Abschlussnoten der Moduleile a) und b) im Verhältnis von 50% : 50% zusammen. Beide Moduleile müssen mit mindestens einer 4,0 bestanden sein. Für die Berechnung der Abschlussnoten und der Modulnote wird die Skala des ZFA verwendet.

Ein nicht bestandener Moduleil muss wiederholt werden, d.h. die Lehrveranstaltung muss erneut belegt und erfolgreich abgeschlossen werden.

Voraussetzungen für die Vergabe von Credits

Zu a) und b)

- Anwesenheit von 75 Prozent

Erfolgreicher Abschluss aller semesterbegleitenden Leistungen (siehe Prüfungsformen)

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]

Titel des Moduls: Mathematik 3
Mathematics 3

Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Mathematik 3 - Anwendungen			Kontaktzeit 105 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr. Peter Eichelsbacher
 Lehrende:

Verwendung des Moduls
 Bachelor Informatik

Lernziele (learning outcomes)

Nach dem erfolgreichen Absolvieren des Moduls

- verfügen die Studierenden über grundlegende Methodenkenntnisse im Bereich der Wahrscheinlichkeitsrechnung und erlernen die Abhängigkeiten von Ereignissen. Insbesondere können sie die Erfolgswahrscheinlichkeit von Algorithmen berechnen
- können die Studierenden die Laufzeit von Algorithmen als Zufallsvariable modellieren und sind in der Lage, Erwartungswert und Varianz dieser Zufallsvariable zu berechnen bzw. mittels geeigneter Schranken akkurat abzuschätzen
- haben die Studierenden die wichtigsten diskreten Verteilungen erlernt und können bei neu auftretenden Problemen klassifizieren, welchen Verteilungen diese angehören
- können die Studierenden Abzählprobleme mithilfe probabilistischer Methoden wie Random Walks oder Monte Carlo Simulation hinreichend gut approximieren
- sind die Studierenden in der Lage, selbstgewählte Hypothesen auf ihre statistische Signifikanz mit geeigneter Konfidenz zu überprüfen
- sind die Studierenden in der Lage, beobachtete experimentelle Daten mithilfe einer Regressionsanalyse geeignet zu klassifizieren

Inhalt

Dieses Modul gibt eine allgemeine Einführung in die Stochastik und behandelt wichtige Gebiete der Wahrscheinlichkeitstheorie, der Statistik und deren Anwendungen.

Folgende Themen werden behandelt:

- Union Bound, Inklusion-Exklusion, Unabhängigkeit, bedingte
- Wahrscheinlichkeiten
- Diskrete Zufallsvariable, Erwartungswert, Momente
- Bernoulli- und Binomialverteilung, Geometrische Verteilung
- Markov, Chebyshev, Chernoff Schranken
- Moment-Erzeugendenfunktion
- Bälle-Urnen Modell, Poisson Verteilung
- Probabilistische Methoden
- Markov Kette, Random Walk
- Monte Carlo Methode, uniformes Samplen
- Einführung in die Statistik
- Punkt- und Intervallschätzung
- Signifikanztests
- Regressionsanalyse

Lehrformen

Hörsaalvorlesung mit Medienunterstützung, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend bereitgestellten Materialien und Aufgaben

Prüfungsformen

Schriftliche Modulabschlussprüfung über 180 Minuten

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung

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

Titel des Moduls: Informatik 3					
Modul-Nr./Code	Credits 8 CP	Workload 240 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen 212002: Informatik 3 - Theoretische Informatik			Kontaktzeit 90 h	Selbststudium 150 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Thomas Zeume					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik B.Sc. IT-Sicherheit / Informationstechnik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • 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. • 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. • haben die Studierenden mit der Turingmaschine ein einfach handhabbares Rechnermodell erlernt, das ihnen fortan als Abstraktion für alle möglichen Rechner dient. • 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 von der Komplexität von Berechnungsproblemen. 					
Inhalt Die Lehrveranstaltung gibt einen systematischen Überblick über die folgenden Themengebiete: <ul style="list-style-type: none"> • Endliche Automaten und reguläre Ausdrücke • Kellerautomaten und kontextfreie Grammatiken • Turingmaschinen und Entscheidbarkeit • Nichtdeterminismus und NP-Vollständigkeitstheorie 					
Lehrformen Hörsaalvorlesung mit Medienunterstützung und Übungen, bei denen die vorgestellten Konzepte und Techniken praktisch umgesetzt werden, teilweise mit Rechnerübungen.					
Prüfungsformen Schriftliche Modulabschlussprüfung (150 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]

Titel des Moduls: Technische Informatik 3					
Modul-Nr./Code	Credits 5 CP	Workload 150 Stunden	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Technische Informatik 3 - Hardwareprogrammierung			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Tim Güneysu Lehrende: Prof. Dr.-Ing. Tim Güneysu					
Verwendung des Moduls Bachelor Informatik					
Lernziele (learning outcomes) 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					
Inhalt 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.					
Lehrformen 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					
Prüfungsformen Schriftliche Modulabschlussprüfung über 120 Minuten					
Voraussetzungen für die Vergabe von Credits 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]					

Titel des Moduls: Software Engineering					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 3	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Software Engineering			Kontaktzeit	Selbststudium 90 h	Gruppengröße 350 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik B.Sc. IT-Sicherheit/Informationstechnik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verfügen die Studierenden über vertiefte Kenntnisse über ausgewählte Aspekte des Softwareentwicklungsprozesses • verfügen die Studierenden über Grundkenntnisse zum Thema Softwarequalität • kennen und verstehen die Studierenden die grundsätzlichen Ziele und Verantwortlichkeiten im Software-Lebenszyklus • kennen und verstehen die Studierenden die verschiedenen Aktivitäten innerhalb des Software-Lebenszyklus und deren Abhängigkeiten • sind die Studierenden in der Lage, die vermittelten Software-Entwurfsmethoden und Entwicklungsprozesse fallspezifisch anzuwenden 					
Inhalt Die Studierenden lernen unterschiedliche Formen von (klassischen und agilen) Vorgehensmodellen in der Softwareentwicklung kennen. Sie lernen Methoden der Anforderungserhebung, des Entwurfs und des Testens kennen und setzen diese in reale Fallbeispiele selbstständig um.					
Lehrformen Die Vorlesung wird als seminaristischer Unterricht mit Medienunterstützung abgehalten, die praktischen Übung am Rechner werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
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) 5/158: 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]					

Titel des Moduls: Betriebssysteme
Operating Systems

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 4	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Betriebssysteme (211005)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 350 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr.-Ing. Timo Hönig
 Lehrende: Prof. Dr.-Ing. Timo Hönig

Verwendung des Moduls

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

Lernziele (learning outcomes)

Nach dem erfolgreichen Absolvieren des Moduls

- erlangen die Studierenden ein solides Grundverständnis von modernen Betriebssystemen, ihrer Funktion und ihrer Implementierung
- 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
- sind die Studierenden in der Lage, bestimmte Aspekte eines Betriebssystems selbst zu designen und diese argumentativ zu verteidigen

Inhalt

In diesem Modul werden die wichtigsten Grundlagen zu Betriebssystemen vorgestellt. Dazu gehören zum Beispiel:

- Betriebssystemkonzepte
- Prozesse und Threads, Interprozesskommunikation
- Scheduling-Mechanismen
- Speicherverwaltung, Speicherabstraktionen, Paging
- Dateisysteme
- Eingabe- und Ausgabeverwaltung
- Algorithmen zur Vermeidung von Deadlocks
- Grundlagen der Sicherheit von Betriebssystemen

In den letzten Wochen der Veranstaltung, abhängig vom verfügbaren Zeitfenster, werden spezielle Themen wie beispielsweise Multimedia-Betriebssysteme, Multiprozessorsysteme und Entwurf von Betriebssystemen, behandelt.

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.

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]

5/165: 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 [PO 22]

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

Titel des Moduls: Computernetze Computer Networks					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 400 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Jörg Schwenk Lehrende: Dr.-Ing. Christian Mainka					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik B.Sc. IT-Sicherheit / Informationstechnik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende die wichtigsten Standards, die das heutige Internet verwendet. • kennen Studierende grundlegende Angriffskonzepte auf Computernetzwerke • verstehen Studierende den Zusammenhang zwischen den einzelnen Schichten eines Computernetzwerks und der darin enthaltenen Protokolle • können Studierende die wichtigsten Netzwerktools für Analysezwecke anwenden 					
Inhalt Die Vorlesung gibt eine Einführung in grundlegenden Protokolle 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.					
Lehrformen Moodle-Unterstützte Hausaufgaben mit praxisnahen, computerunterstützten Übungen. Tool-der-Woche: Vorstellung, Einarbeitung, und Verwendung von Netzwerkrelevanten Computeranalysetools.					
Prüfungsformen 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)					

Titel des Moduls: Einführung in die künstliche Intelligenz Introduction to Artificial Intelligence					
Modul-Nr./Code	Credits 5 CP	Workload	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Introduction to Artificial Intelligence (211045)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 250 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott, Prof. Dr. Tobias Glasmachers, Prof. Dr. Sen Cheng, Prof. Dr. Gregor Schöner, Prof. Dr. Maribel Acosta, Prof. Dr. Christian Straßer					
Verwendung des Moduls B.Sc. Informatik (Pflichtmodul) B.Sc. Angewandte Informatik (Pflichtmodul) B.Sc. IT-Sicherheit/Informationstechnik (Wahlpflichtmodul)					
Lernziele (learning outcomes) After successful completion of this course, students will be able to <ul style="list-style-type: none"> • summarize a number of fundamental methods in artificial intelligence, • explain their mathematical basis and algorithmic nature, • apply them to simple problems, • decide which methods are suitable for which problems, and • communicate about the all that in English. 					
Inhalt 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.					
Lehrformen This course is given with the flipped/inverted classroom concept. The students work through online material beforehand and this will then be deepened in the contact sessions, which will be used for an interactive exchange between students and with the lecturer in a flexible format.					
Prüfungsformen Written module final exam (120 minutes)					
Voraussetzungen für die Vergabe von Credits passed written exam					
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]					

Titel des Moduls: Datenbanksysteme Database Systems					
Modul-Nr./Code	Credits 7 CP	Workload 210 h	Semester 4	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Datenbanksysteme (211008)			Kontaktzeit 60 h	Selbststudium 150 h	Gruppengröße 250 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Jun.-Prof. Dr.-Ing. Maribel Acosta Deibe Lehrende: Jun.-Prof. Dr.-Ing. Maribel Acosta Deibe					
Verwendung des Moduls B.Sc. Informatik [PO 22] B.Sc. Angewandte Informatik [PO 22]					
Lernziele (learning outcomes) Nach dem erfolgreichen Absolvieren des Moduls <ul style="list-style-type: none"> • erlangen die Studierenden ein Grundverständnis von modernen Datenbanksystemen, ihrer Funktion und ihrer Implementierung • haben die Studierenden Datenmodellierungstechniken erlernt • haben die Studierenden die Semantik und die Syntax des Entity-Relationships Modells kennengelernt • kennen die Studenten das relational Datenbankmodell und die Relationale Algebra • kennen die Studierenden Anfragesprachen (z.B. SQL) und können diese nutzen • verstehen die Studierenden die Konzepte von Transaktion und Fehlerbehandlung • haben die Studierenden unterschiedliche Datenbankmanagementsysteme kennengelernt • sind die Studierenden in der Lage, neue Datenbanken zu modellieren und zu implementieren • haben die Studenten Kenntnisse über die Prozesse hinter einer Datenbankanfrage und wie diese optimiert werden kann 					
Inhalt 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"> • Einführung in Datenbanksysteme • Entity-Relationship Modell und Verbesserungen • das relational Datenbankmodell • Relationale Algebra und Kalkül • Die Relationale Anfragesprache SQL • Datenbankprogrammierung • physische Datenorganisation • Anfragebearbeitung und Optimierung • Transaktionsverwaltung und Fehlerbehandlung 					
Lehrformen In der wöchentlichen Vorlesung werden die Lerninhalte theoretisch vermittelt. In der unterstehenden 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]

Titel des Moduls: Software Engineering Praktikum
Software Engineering Lab

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Software Engineering Lab (211500)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 5 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik					
Lernziele (learning outcomes) Nach dem Modulabschluss <ul style="list-style-type: none"> • können die Studierenden agil arbeiten nach SCRUM • wissen Studierende, wie man kleine Software-Projekte plant und können diese in Java-Android umsetzen • können Studierende ihre eigenen Ergebnisse in angemessener Form präsentieren 					
Inhalt 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.					
Lehrformen Agiles Arbeiten in Projektgruppen					
Prüfungsformen Projektarbeit (semesterbegleitend) mit Zwischenmeetings, Abgaben und Abschlusspräsentation					
Voraussetzungen für die Vergabe von Credits erfolgreich abgeschlossene Projektarbeit mit Abschlusspräsentation.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 5/158: B.Sc. Informatik [PO 22] 5/168: B.Sc. Angewandte Informatik [PO 22]					

Titel des Moduls: Algorithmenparadigmen					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Algorithmenparadigmen (211043)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 40 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Maïke Buchin Lehrende: Prof. Dr. Maïke Buchin					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende eine Reihe von Algorithmenparadigmen • können Studierende basierend auf den Paradigmen effiziente Algorithmen für Probleme entwickeln • verstehen Studierende die Vor- und Nachteile unterschiedlicher Paradigmen 					
Inhalt In der Vorlesung betrachten wir unterschiedliche Algorithmenparadigmen, also Schemata zum Entwurf von effizienten Algorithmen. Dazu betrachten wir zunächst die bereits bekannten Paradigma 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.					
Lehrformen Hörsaalvorlesung mit Medienunterstützung sowie Tutorien als seminaristischer Unterricht					
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) 5/170: 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: Autonomous Vehicles and Artificial Intelligence					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 5	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Autonomous Vehicles and Artificial Intelligence			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 25 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger, Dr. Sven Peldszus					
Verwendung des Moduls B.Sc. Informatik B.Sc. IT-Sicherheit B.Sc. Angewandte Informatik M.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit/Informationstechnik					
Lernziele (learning outcomes) <ul style="list-style-type: none"> • Understanding requirements on autonomous vehicles • Understanding the architecture of autonomous vehicles • Ability to build a self-driving car with ROS2 • Understanding and applying quality assurance for autonomous vehicles 					
Inhalt Autonomous driving is the future of individual mobility and all major manufacturers are working on fully autonomous vehicles. While there are robust and good solutions for the individual problems in autonomous driving, the main challenge lies in their integration. Altogether, an autonomous vehicle's software is the biggest problem. Therefore, the key in self-driving vehicles is about getting the software right. In this course, we will investigate the different aspects of self-driving vehicles as well as the importance and application of artificial intelligence in this domain. The course will primarily focus on the following topics: <ul style="list-style-type: none"> • Requirements on autonomous vehicles • Architecture of autonomous vehicles • Operation systems and frameworks for robotic systems • Specification and Implementation of autonomous vehicles based on ROS2 • Artificial intelligence for autonomous vehicles • Simulation of autonomous vehicles &#8729; Localization and perception • Mission planning • Quality assurance for autonomous vehicles In the course's lecture, we provide the required theoretical background and practically apply the course's content in exercises by building a self-driving robot. 					
Lehrformen Vorlesung mit Übungen					
Prüfungsformen Mündliche Prüfung					
Voraussetzungen für die Vergabe von Credits 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 22]

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

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

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

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

5/ 97: M.Sc. Informatik

5/105: M.Sc. Angewandte Informatik

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

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

Titel des Moduls: Digitale Forensik					
Modul-Nr./Code	Credits 5 CP	Workload	Semester	Turnus Wintersemester	Dauer Semester
Lehrveranstaltungen			Kontaktzeit	Selbststudium	Gruppengröße Studierende
Unterrichtssprache			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Lehrende:					
Verwendung des Moduls					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen					
Prüfungsformen					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)					

Titel des Moduls: Einführung in die Kryptographie 1

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 1	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Einführung in die Kryptographie 1 (212010)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 300 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen 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					
Lernziele (learning outcomes) 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 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 symmetrische und asymmetrische Verfahren und Algorithmen werden vorgestellt und an praxisrelevanten Beispielen erläutert. Die Vorlesung lässt sich in zwei Teile gliedern: Die Funktionsweise der symmetrischen Kryptographie einschließlich der Beschreibung historisch bedeutender symmetrischer Verschlüsselungsverfahren (Caesar Chiffre, Affine Chiffre) und aktueller symmetrischer Verfahren (Data Encryption Standard, Advanced Encryption Standard, Stromchiffren, One Time Pad) werden im ersten Teil behandelt. Der zweite Teil besteht aus einer Einleitung zu asymmetrischen Verfahren und einem ihrer wichtigsten Stellvertretern (RSA). Hierzu wird eine Einführung der Grundlagen der Zahlentheorie durchgeführt, um ein grundlegendes Verständnis der Verfahren sicherzustellen (u.a. Ringe ganzer Zahlen, Gruppen, Körper, diskrete Logarithmen, euklidischer Algorithmus). Nichtsdestotrotz liegt der Schwerpunkt auf der algorithmischen Einführung des asymmetrischen Verfahrens.					
Lehrformen Vorlesung mit Übung					
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 22]

/ B.Sc. IT-Sicherheit [PO 22]

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

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

Titel des Moduls: Einführung in die Kryptographie 2

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

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-Schlusselaustausch 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)

/: 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]

Titel des Moduls: Game Development					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Game Development (211001)			Kontaktzeit 30 h	Selbststudium 150 h	Gruppengröße Studierende
Unterrichtssprache Deutsch, Kursmaterial auf Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: M. Sc. Daniel Vonk					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verstehen die Studierenden Grundlagen der objektorientierten Programmierung mit C# im Rahmen der Unity-Engine, • haben die Studierenden ein umfassendes Wissen über den Bereich der Spieleentwicklung erworben und kennen moderne Tools sowie aktuelle Methoden der 2D- und 3D-Entwicklung, • können die Studierenden praxisnahe Problemstellungen der Softwareentwicklung analysieren und eigenständig lösen, • können die Studierenden Projekte im Bereich der Spieleentwicklung definieren und fachgerecht umsetzen 					
Inhalt Die Veranstaltung bietet einen umfangreichen Einblick in viele Bereiche der Spieleentwicklung. Dazu gehören: <ul style="list-style-type: none"> • Grundlagenwissen (Spiele-Engines, moderne Softwaretools, Projektmanagement) • C#-Grundlagen (Syntax, Datentypen, Operatoren, Kontrollstrukturen) • Benutzerinteraktion (In-/Output mit Tastatur sowie Controller, User Interfaces) • Gameplay (Bewegen von Spielobjekten, Kamerasteuerung, Game Loop und Framerates) • Physik (Rigidbody, Collider, Trigger) • Assets (Import von Bildern, Audio und 3D-Modellen sowie Erstellung von Animationen) • Grafik (Texturen, Partikeleffekte, Beleuchtung, Post-Processing) • 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. 					
Lehrformen Online-Videos und wöchentliche Hörsaalübungen					
Prüfungsformen 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: Information Theory					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Information Theory			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter					
Verwendung des Moduls B.Sc. Informatik B.Sc. IT-Sicherheit M.Sc. Informatik M.Sc. IT-Sicherheit / Informationstechnik M.Sc. Angewandte Informatik					
Lernziele (learning outcomes) You will learn fundamental concepts, algorithms, and results in information theory. After successful completion of this course, you will know the mathematical model of information theory, how to design and analyze algorithms for a variety of information processing tasks, and how to implement them in Python. You will have independently read about a topic in information theory and presented it to your peers. You will be prepared for an advanced course or a research or thesis project in this area. Please see the course homepage for a precise list of learning objectives.					
Inhalt This course will give an introduction to information theory – the mathematical theory of information. Ever since its inception, information theory has had a profound impact on society. It underpins important technological developments, from reliable memories to mobile phone standards, and its versatile mathematical toolbox has found use in computer science, machine learning, physics, electrical engineering, mathematics, and many other disciplines. Starting from probability theory, we will discuss how to mathematically model information sources and communication channels, how to optimally compress information, and how to design error-correcting codes that allow us to reliably communicate over noisy communication channels. We will also see how techniques used in information theory can be applied more generally to make predictions from noisy data.					
Tentative syllabus: - Welcome, Introduction to Information Theory - Probability Theory Refresher - Numerical Random Variables, Convexity and Concavity, Entropy - Symbol Codes: Lossless Compression, Huffman Algorithm - Block Codes: Shannon's Source Coding Theorem, its Proof, and Variations - Stream Codes: Lempel-Ziv Algorithm - Stream Codes: Arithmetic Coding					

- Joint Entropies & Communication over Noisy Channels
- Shannon's Noisy Coding Theorem
- Proof of the Noisy Coding Theorem
- Proof of the Converse, Shannon's Theory vs Practice
- Reed-Solomon Codes
- Message Passing for Decoding and Inference, Outlook
- Student Presentations

Please see the course homepage https://qi.rub.de/it_ss23 for more information.

Lehrformen

Lecture with Exercise

Prüfungsformen

Wird noch nachgeliefert

Voraussetzungen für die Vergabe von Credits

Passed Exam.

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]

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

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

5/97: M.Sc. Informatik

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

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

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Logik in der Informatik					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 5	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen 212013: Vorlesung + Übung			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 150 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Thomas Zeume					
Verwendung des Moduls B.Sc. Angewandte Informatik B.Sc Informatik					
Lernziele (learning outcomes) 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.					
Inhalt 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.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche Modulabschlussprüfung					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 5/170: B.Sc. Angewandte Informatik [PO 20]					

Titel des Moduls: Model Checking					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Model Checking			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Dr. Nils Vortmeier Marko Schmellenkamp					
Verwendung des Moduls B.Sc. Informatik M.Sc. Informatik M.Sc. Angewandte Informatik B.Sc. IT-Sicherheit/Informationstechnik M.Sc. Mathematik					
Lernziele (learning outcomes) Die Studierenden lernen wie sich verteilte Systeme durch Transitionssysteme modellieren und Eigenschaften in logischen Spezifikationsprachen wie LTL und CTL spezifizieren lassen. Sie sollen elementare Algorithmen zur Überprüfung von Eigenschaften in Transitionssystemen kennenlernen. Sie sollen ein Verständnis für die Möglichkeiten und Grenzen des Model Checking entwickeln, und in die Lage versetzt werden, sich eigenständig mit fortgeschrittenen Methoden des Model Checkings auseinanderzusetzen.					
Inhalt Wie kann die Korrektheit von Software und Hardware formal überprüft werden? Im Model Checking werden Software- und Hardware-Module durch Transitionssysteme formalisiert; gewünschte Eigenschaften mit Hilfe logischer Formalismen formal beschrieben; und mit Hilfe von Algorithmen automatisiert überprüft, ob ein Transitionssystem eine formal spezifizierte Eigenschaft besitzt. In dieser Veranstaltung werden die theoretischen Grundlagen des Model Checkings vermittelt, mit einem Fokus auf logik-basierten Spezifikationsprachen. Die Spezifikationsprachen LTL und CTL werden eingeführt, ihre Ausdrucksstärke untersucht, und die wichtigsten algorithmischen Ansätze für das Model Checking vorgestellt.					
Lehrformen Vorlesung mit Übungen					
Prüfungsformen Abschlussprüfung; mündliche Prüfung (20-30min) oder schriftliche Klausur (120min) in Abhängigkeit der Teilnehmerzahl					
Voraussetzungen für die Vergabe von Credits Erfolgreiches Bestehen der Modulprüfung.					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 5/170: B.Sc. Informatik [PO 22] 5/158: B.Sc. Informatik [PO 20] 5/97: M.Sc. Informatik 5/105: M.Sc. Angewandte Informatik					

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

5/149: B.Sc. IT-Sicherheit/Informationstechnik [PO 22]

Titel des Moduls: Nebenläufige Programmierung

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 6	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Nebenläufige Programmierung (211012)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende
Modulbeauftragte/r: Dr.-Ing. Doga Arinir (Lehrauftrag)
Lehrende: Dr.-Ing. Doga Arinir

Verwendung des Moduls

B.Sc. Informatik

B.Sc. Angewandte Informatik

Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- haben die Studierenden grundlegende Fähigkeiten und Techniken erworben, um nebenläufige Programme sicher entwickeln zu können
- kennen die Studierenden softwaretechnische Entwurfsmuster, welche bekannte Probleme bei nebenläufigen Programmen, wie zum Beispiel die Verklemmung, vermeiden lassen
- können die Studierenden die Performanz von Programmen durch den Einsatz der nebenläufigen Programmierung verbessern
- sind die Studierenden in der Lage, bestehende Programme zu analysieren und mögliche Fehler zu erkennen
- können die Studierenden die Sprachmerkmale und Schnittstellen von JAVA für die nebenläufige Programmierung sicher anwenden

Inhalt

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

- Anwendungen vs. Prozesse
- Programme und ihre Ausführung
- Vorteile und Probleme von nebenläufigen Programmen (Verbesserung der Performanz, Synchronisation, Realisierung kritischer Abschnitte, Monitore, Lebendigkeit, Verklemmungen)

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]

Titel des Moduls: Network Planing

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Network Planing			Kontaktzeit 45 h	Selbststudium 105	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Aydin Sezgin

Lehrende: Prof. Aydin Sezgin

Verwendung des Moduls

Bachelor Informatik

Lernziele (learning outcomes)

Die Studierenden beherrschen die Behandlung zentraler Aspekte der Linearen Optimierung. Dies sind:

- die Modellierung von Problemen im Bereich der Informationstechnik (z.B. Leistungsallokation) sowie im Alltag (z.B. Rucksackproblem, Sudoku, Ernährung) als lineare Optimierungsprobleme
- die Dualität sowie notwendige und hinreichende Bedingungen
- Verfahren, die zur effizienten Bestimmung von Lösungen führen

Inhalt

In vielen technischen (aber auch nichttechnischen) Bereichen werden Lösungen für Probleme gesucht, bei denen auch immer gewisse Vorgaben oder Nebenbedingungen erfüllt werden müssen. Die Optimierung dient hierbei als systematisches Werkzeug zur effizienten Lösungsbestimmung. Der Anwendungsfokus der Vorlesung ist in der Netzwerk-Planung wie Interferenz-Management, Frequenz- und Nutzerzuweisungen, Positionierung von Basisstationen sowie Routing.

1. Einleitung und Überblick
 - Motivation, Formulierung von linearen Problemen, Varianten, Beispiele, stückweise lineare Zielfunktionen
 - Graphische Darstellung und Lösung
 - Lineare Algebra: Überblick und Notation
2. Geometrie der linearen Optimierung
 - Konvexe Mengen, Polyhedra, Extrempunkte
3. Die Simplex-Methode
 - Optimalitätsbedingungen, Entwicklung, Implementierung
4. Dualitätstheorie
 - Motivation, Duales Problem, Dualitätstheorem
5. Spieltheorie
6. Sensitivitätsanalyse (Lokale)
7. Netzwerk-Fluss-Probleme
 - Formulierung, Probleme: Kürzester Pfad/Maximaler Fluss, Netzwerk-Simplex Algorithmus
8. Innere-Punkt-Methoden
 - Affiner Skalierungsalgorithmus
9. Ganzzahlige Optimierung
 - Formulierung
 - Methoden: Branch and bound, cutting plane
10. Anwendungen

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Es gibt jeweils 4 Übungsblätter mit theoretischen Teilaufgaben mit insgesamt 15 Punkten und zusätzlich 4 Programmieraufgaben mit je 10 Punkten. Die Prüfung ist bestanden, wenn 30 Punkte in den theoretischen Aufgaben und 20 Punkte in den Programmieraufgaben erreicht werden.

Voraussetzungen für die Vergabe von Credits

Erfolgreiches Bestehen der 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]

Titel des Moduls: Proofs are programs

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 40 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		

Modulbeauftragte/r und hauptamtlich Lehrende

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

Verwendung des Moduls

o Bachelor Informatik

Lernziele (learning outcomes)

After successful completion of this course, students will be able to

- develop purely functional programs using recursive functions on numbers, lists, maps, and various kinds of trees, including the abstract syntax trees of programs;
- use functional programming concepts such as type polymorphism and higher-order functions, which are increasingly becoming mainstream;
- formally state and prove theorems in the Coq proof assistant;
- apply different proof techniques in Coq (e.g. equational reasoning, contradiction, case analysis, induction on natural numbers, structural induction, proof automation);
- define new inductive types and relations in Coq and prove statements about them;
- understand the connection between constructive logics and typed functional programming that is at the heart of Coq, in which propositions are types and proofs are programs;
- understand how the syntax and semantics of simple imperative programs can be formally defined in Coq and how to prove theorems about such programs and languages.

Inhalt

Complex mathematical proofs on paper are difficult to write, check, and maintain. This holds not only for interesting proofs in mathematics, but also for complex formal proofs about interesting programs. For this reason, machine-checked proofs created with the help of interactive tools called proof assistants are gaining increased traction in academia and industry. Proof assistants have been used to prove the correctness and security of realistic compilers, operating systems, cryptographic libraries, or smart contracts, and also to construct machine-checked proofs for challenging mathematical results such as the four color theorem, the odd-order theorem (Feit–Thompson), or the construction of perfectoid spaces.

This course introduces the Coq proof assistant and explains how to use it to prove properties about functional programs and inductive relations. The Coq proof assistant enables us to program formal proofs interactively and it machine-checks the correctness of the proofs along the way. The design of the Coq proof assistant itself exploits a beautiful connection between programs in typed functional programming languages and proofs in constructive logics, which is known as the Curry-Howard Correspondence. This deep connection between programs and proofs should make this course interesting to both computer scientists and mathematicians. For computer scientists the goal is to demystify proofs as just programs in an elegant programming language, for which the course provides a gentle introduction. For mathematicians this course serves as an introduction to functional programming and also to the idea that proofs are not only a way to convince a human reader, but they can actually be fully formalized in a proof assistant like Coq and automatically checked by a computer.

This hands-on course is based on the Logical Foundations online textbook, which is itself formalized and machine-checked in the Coq proof assistant. The many exercises in each book chapter are to be solved weekly mostly in Coq, from easy exercises allowing the students to practice concepts from the lecture, building incrementally to slightly more interesting programs and proofs and also to various optional challenges. Finally, this course serves as the base for a more advanced course on “Foundations of Programming Languages, Verification,

and Security”.

Lehrformen

This course consists of lectures and weekly exercises, in which the students will solve problems using the Coq proof assistant for which they can get help from a tutor.

Prüfungsformen

Written final exam (90 minutes). We also plan to have a midterm exam that helps students practice for the final exam, but only counts for bonus points.

Voraussetzungen für die Vergabe von Credits

Passing the final exam (60% of the grade) and timely submission of solutions for the weekly Coq exercise sheets (40% of the grade).

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

Titel des Moduls: Quantum Information and Computation					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Quantum Information and Computation			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache German or Englisch (depends on audience)			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter					
Verwendung des Moduls B.Sc. Informatik B.Sc. IT-Sicherheit/Informationstechnik M.Sc. Angewandte Informatik					
Lernziele (learning outcomes) You will learn fundamental concepts, algorithms, and results in quantum information and computation. After successful completion of this course, you will know the theoretical model of quantum information and computation, how to generalize computer science concepts to the quantum setting, how to design and analyze quantum algorithms and protocols for a variety of computational problems, and how to prove complexity theoretic lower bounds. You will be prepared for an advanced course or a research or thesis project in this area.					
Inhalt This course will give an introduction to quantum information and quantum computation from the perspective of theoretical computer science. Topics to be covered will likely include: <ul style="list-style-type: none"> • Fundamentals of quantum computing: quantum bits, states and operations • The power of quantum entanglement: nonlocal games • Entanglement as a resource: superdense coding and teleportation • Quantum circuit model of computation • Quantum computing with oracles: Deutsch-Jozsa, Bernstein-Vazirani, Simon • Quantum Fourier transform and phase estimation • Shor's factoring algorithm • Grover's search algorithm and beyond: how to solve SAT on a quantum computer? • From no cloning to quantum money: a peek at quantum cryptography <p>The course should be of interest to students of computer science, mathematics, physics, and related disciplines. Students interested in a BSc or MSc project in quantum information, computing, cryptography, etc. are particularly encouraged to participate.</p>					
Lehrformen Lecture with Exercise					
Prüfungsformen Written Exam (120 Minutes)					
Voraussetzungen für die Vergabe von Credits Passed written Exam.					
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]					

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

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

5/105: M.Sc. Angewandte Informatik

Titel des Moduls: Statistisches Lernen und Data Mining					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Statistisches Lernen und Data Mining			Kontaktzeit 60 h	Selbststudium 90	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Johannes Lederer Lehrende: Prof. Dr. Johannes Lederer					
Verwendung des Moduls B.Sc. Informatik M.Sc. Informatik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls - kennen die Studierenden Standardmethoden der Datenanalyse - verstehen die Studierenden, wann welche Methoden passend sind - sind die Studierenden in der Lage, die Methoden anzuwenden - können die Studierenden die Ergebnisse interpretieren)					
Inhalt In dieser Lehrveranstaltung werden die grundlegenden Methoden der Datenanalyse eingeführt. Dabei werden verschiedene Datentypen berücksichtigt, insbesondere Regressionsdaten und Klassifikationsdaten. Immer werden auch die zu Grunde liegenden statistischen Modelle besprochen. Ebenfalls werden mögliche Anwendungen sowohl im Unterricht als auch in Computer-Übungen vorgestellt. 105 Ziel ist es, den gesamten Verlauf einfacher Datenanalysen zu vermitteln: Datenaufbereitung, statistische Modellbildung, Auswahl einer Methode, Implementierung der Methode, Visualisierung der Resultate und Interpretation.					
Lehrformen Hörsaalvorlesung mit Medienunterstützung, insbesondere Datenanalysen mit dem Computer, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend bereitgestellten Materialien und Aufgaben					
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]					

Titel des Moduls: Systemsicherheit System Security					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Systemsicherheit			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Ghassan Karame Lehrende: Prof. Dr. Ghassan Karame					
Verwendung des Moduls B.Sc. IT-Sicherheit / Informationstechnik B.Sc. Informatik M.Sc. Angewandte Informatik M.Sc. IT-Sicherheit / Netze und Systeme					
Lernziele (learning outcomes) At the end of this course, students will be able to <ul style="list-style-type: none"> • classify and describe vulnerabilities and protection mechanisms of popular systems and protocols, and • analyze / reason about basic protection mechanisms for modern OSs, software, and hardware systems. Students will also develop the ability to reason about the security of a given protocol and independently develop appropriate security defenses and security models. 					
Inhalt While clearly beneficial, the large-scale deployment of online services has resulted in the increase of security threats against existing services. As the size of the global network grows, the incentives of attackers to abuse the operation of online applications also increase and their advantage in mounting successful attacks becomes considerable. These cyber-attacks often target the resources, availability, and operation of online services. With an increasing number of services relying on online resources, integrating proper security measures therefore becomes integral to ensure the correct functioning of every online service. In this course, we discuss important theoretical and analytical aspects in system security. The focus of the course is to understand basic attack strategies on modern systems and platforms, with a focus on side-channel attacks, software-based attacks, malware analysis, as well as software-based defenses (e.g., address space randomization and non-executable memory) and hardware-based defenses (e.g., using TPMs and TEEs). Other topics of the course include analyzing the security of modern cryptocurrencies and ML platforms, and similar aspects in system security. An integral part of this course are exercises and homeworks, which aim to deepen the understanding of the material with practical examples.					
Lehrformen Lecture with Exercise					
Prüfungsformen Written Exam (120 Minutes)					
Voraussetzungen für die Vergabe von Credits Passed written Exam					

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]

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

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

5/105: M.Sc. Angewandte Informatik

5/99: M.Sc. IT-Sicherheit / Netze und Systeme [PO 22]

5/96: M.Sc. IT-Sicherheit / Netze und Systeme [PO 20]

Titel des Moduls: Vertiefungspraktikum Informatik					
Modul-Nr./Code	Credits 3 CP	Workload 90 h	Semester 5	Turnus Wintersemester oder Sommersemester	Dauer 1 Semester
Lehrveranstaltungen <ul style="list-style-type: none"> • Praktikum Systemsoftware-Technik • Praktikum Implementing Post Quantum Standards and Challenges • Praktikum IDE Plugin Development • Android App Evolution • Introduction to Python • Praktische Kryptanalyse von symmetrischen Chiffren 			Kontaktzeit Je nach Veranstaltungswahl	Selbststudium Abhängig von Praktikumswahl	Gruppengröße Studierende
Unterrichtssprache Abhängig von Praktikumswahl: Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan der Angewandten Informatik Lehrende: siehe jeweiligen Praktikumseintrag					
Verwendung des Moduls B.Sc. Informatik					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • haben Studierende Ihre Fähigkeiten im Programmieren vertieft und erweitert • je nach gewählten Programmierpraktikum können noch weitere Lernziele dazu kommen 					
Inhalt Es werden aktuell Praktika zu folgenden Themen angeboten: <ul style="list-style-type: none"> • Praktikum Systemsoftware-Technik • Praktikum Implementing Post Quantum Standards and Challenges • Praktikum IDE Plugin Development • Android App Evolution • Introduction to Python • Praktische Kryptanalyse von symmetrischen Chiffren Weiterführende Informationen zu den jeweiligen Praktika finden Sie im Vorlesungsverzeichnis.					
Lehrformen Praktikum im Block oder als semesterbegleitende Veranstaltung.					
Prüfungsformen Praktikum					
Voraussetzungen für die Vergabe von Credits Erfolgreiche Teilnahme am Praktikum					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) Informatik PO20 : 3/165 Informatik PO22 : unbenotet					

Titel des Moduls: Vertiefungsseminar Informatik

Modul-Nr./Code	Credits 3 CP	Workload 90 h	Semester 5	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen <ul style="list-style-type: none">• Seminar Satisfiability (211117)• Seminar zu Approximationsalgorithmen (211118)• Seminar From Biological to Artificial Neural Networks (211131)• Seminar über Grenzen in der theoretischen Informatik (211117)• Seminar Reinforcement Learning (212119)• Seminar Knowledge Graphs (212113)• Machine Learning Applications (212108)• Seminar Ressourceneffiziente Systemsoftware (212111)• Seminar Implementation Security (212126)• Seminar zur symmetrischen Kryptographie (212118)• Seminar Modern Programming Languages (212115)• Seminar Quantum Algorithms (211119)			Kontaktzeit 30 h	Selbststudium 60 h	Gruppengröße Studierende
Unterrichtssprache Je nach Seminarwahl: Deutsch oder Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan Informatik Lehrende: siehe jeweilige Seminare					
Verwendung des Moduls <ul style="list-style-type: none">• Bachelor Informatik (PO 22)• Bachelor Informatik (PO 20)					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none">• verfügen Studierende über vertiefte wissenschaftliche Kenntnisse in dem ausgewählten Seminarthema• haben Studierende das halten eines wissenschaftlichen Vortrags praktisch eingeübt und können Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln• können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen					
Inhalt 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.					
Lehrformen Seminar					

Prüfungsformen

Seminarvortrag

Voraussetzungen für die Vergabe von Credits

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/: B.Sc. Informatik [PO 22]

3/165: B.Sc. Informatik [PO 20]

Titel des Moduls: Web-Engineering					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 4	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Web-Engineering (128968 + 128969)			Kontaktzeit 60 h	Selbststudium	Gruppengröße 200 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Markus König Lehrende: Prof. Dr.-Ing. Markus König Stephan Embers, M.Sc.					
Verwendung des Moduls B.Sc. Angewandte Informatik B.Sc. Informatik M.Sc. IT-Sicherheit/Informationstechnik					
Lernziele (learning outcomes) 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"> • kennen Studierende gängige Konzepte der Web-Entwicklung in den Aspekten Präsentation, Transport und Bereitstellung von Daten • beherrschen Studierende grundlegende Fähigkeiten in Webseitendarstellung, dynamischen Web-Anwendungen und modernen Services (Node.js) 					
Inhalt Im Rahmen des Modules 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"> • Einführung in clientseitige Web-Entwicklung: HTML, CSS, JavaScript, Web Components • Transportverfahren und deren Nutzung: Representational State Transfer (REST), Asynchronous JavaScript und XML (AJAX) • Serverseitige Entwicklung mit Node.js und weiterführende Technologien 					
Lehrformen Hörsaalvorlesung mit Medienunterstützung, 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]

Titel des Moduls: Freies Wahlmodul					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Jedes Semester	Dauer 1 Semester
Lehrveranstaltungen			Kontaktzeit abhängig von Veranstaltungswahl	Selbststudium Je nach Veranstaltungswahl	Gruppengröße Studierende
Unterrichtssprache Je nach Veranstaltungswahl			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studienfachberatung Informatik Lehrende:					
Verwendung des Moduls					
Lernziele (learning outcomes) Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den freien Wahlfächern					
Inhalt Studierende müssen Veranstaltungen im Gesamtumfang von 5 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 ihre Soft Skills zu erweitern. Z.B. Die freien Wahlfächer erweitern die Soft Skills. 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.					
Lehrformen					
Prüfungsformen abhängig von Veranstaltungswahl					
Voraussetzungen für die Vergabe von Credits abhängig von Veranstaltungswahl					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 5/180: B.Sc. Informatik [PO 22] - unbenotet					

Titel des Moduls: Praktische Ausbildung					
Modul-Nr./Code	Credits 15 CP	Workload 300 h - 450 h (10 -15 CP)	Semester 5	Turnus Jedes Winter und Sommersemester	Dauer 1 Semester
Lehrveranstaltungen <ul style="list-style-type: none"> • Industriepraktikum [10 - 15 CP] • Software-Projekt [10 CP] • Forschungsprojekt [10 CP] 			Kontaktzeit 15 h	Selbststudium 270 h	Gruppengröße Studierende
Unterrichtssprache Deutsch / Englisch			Teilnahmevoraussetzungen Abgeschlossenes Modul „Grundlagenpraktikum“		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan*in Informatik Lehrende:					
Verwendung des Moduls Bachelor Informatik					
Lernziele (learning outcomes) Nach erfolgreichem Abschluss des Moduls <ul style="list-style-type: none"> • sind die Studierenden in der Lage, das erlernte Fachwissen anzuwenden • haben die Studierenden die Software-Entwicklungskompetenz maßgeblich ausgebaut, insbesondere im Hinblick auf größere Software-Systeme • haben die Studierenden zusätzliche Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung erworben • können die Studierenden eigene Lösungsstrategien erarbeiten • haben die Studierenden die Fähigkeit zur Arbeitsteilung und Zusammenarbeit im Team verbessert (Teamfähigkeit und Projektorganisation) • haben die Studierenden die Kompetenz im Hinblick auf die Dokumentation von der erstellten Software, aber auch im Hinblick auf die Dokumentation der eigenen Projektarbeit gefestigt • haben die Studierenden die Kompetenz im Hinblick auf die Präsentation von Projektergebnissen verbessert • haben die Studierenden Erfahrung im Bewerbungsprozess gesammelt und sind auf das Berufsleben gut vorbereitet 					
Inhalt Während der praktischen Ausbildung sollen verschiedene Arbeitsgebiete, die im Zusammenhang mit der späteren Tätigkeit einer Informatikerin bzw. eines Informatikers stehen, bearbeitet werden. Im Vordergrund soll die Entwicklung größerer Software- oder anderer IT-Systeme stehen.					
Lehrformen Projektarbeit					
Prüfungsformen Projektarbeit mit tabellarischer Dokumentation					
Voraussetzungen für die Vergabe von Credits Erfolgreich abgeschlossenes Projekt und positiv bewertete abgegebene Dokumentation					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) Dieses Modul wird mit einem Leistungsnachweis (erfolgreich bestanden/nicht bestanden) abgeschlossen und wird nicht benotet.					

Titel des Moduls: Abschlussarbeit					
Modul-Nr./Code	Credits 15 CP	Workload 450 h	Semester	Turnus Jedes Semester	Dauer Semester
Lehrveranstaltungen <ul style="list-style-type: none"> a) Bachelor-Thesis (12 CP) b) Colloquium (3 CP) 			Kontaktzeit 15 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Mindestens 135 absolvierte Leistungspunkte mit abgeschlossenen Module der ersten vier Semester		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Modulbeauftragte/r: Studiendekan*in Informatik Erstbetreuer*in: Jede/r am Studiengang beteiligte Hochschullehrer*in (s. Prüfungsordnung für die Regularien) Lehrende:					
Verwendung des Moduls <ul style="list-style-type: none"> Bachelor Informatik [PO 22] Bachelor Informatik [PO 20] 					
Lernziele (learning outcomes) Die Bachelorarbeit soll zeigen, dass die oder der Studierende in der Lage ist, innerhalb einer vorgegebenen Frist eine anspruchsvolle Fragestellung der Informatik unter Anwendung der im Bachelorstudium erworbenen Methoden selbstständig zu bearbeiten. Darüber hinaus wird der Erwerb von Grundkenntnissen der wissenschaftlichen Arbeit einschließlich der Projektorganisation sowie die Präsentation der erarbeiteten Ergebnisse erwartet. Während der Bachelorarbeit werden die folgenden Kompetenzen erworben bzw. ausgebaut: <ul style="list-style-type: none"> Vertieftes Wissen im Bereich der bearbeiteten Aufgabenstellung Wissenschaftliches Arbeiten und Schreiben Projekt- und Zeitmanagement Präsentation wissenschaftlicher Ergebnisse Rhetorik und sprachliche Kompetenz Fächerübergreifendes Denken und Arbeiten 					
Inhalt a) Bearbeitung und Lösung einer wissenschaftlichen Aufgabe im Bereich der Informatik unter Anleitung. Die im Bachelorstudium erworbenen Kenntnisse, Kompetenzen und Methoden sollen angewendet werden. Die Ergebnisse der Arbeit sind schriftlich zu verfassen. b) Im Anschluss an die Bearbeitung der Bachelorarbeit werden die Ergebnisse in Form eines Kolloquium-Vortrags präsentiert. Als Vorbereitung müssen die Studierenden mindestens fünf Kolloquium-Vorträge anderer Studierenden besuchen und kritisch mitdiskutieren. Außerdem werden sie dazu eingeladen und motiviert, Vorträge des wissenschaftlichen Personals und anderer Gastwissenschaftler zu besuchen und an den Diskussionen aktiv teilzunehmen.					
Lehrformen Projektarbeit					
Prüfungsformen Schriftliche Ausarbeitung der gestellte Aufgabe und Präsentation der Ergebnisse im Kolloquium					
Voraussetzungen für die Vergabe von Credits Positive Bewertung der Bachelorarbeit und des Kolloquiums sowie Teilnahme an anderen wissenschaftlichen Vorträgen					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) <ul style="list-style-type: none"> Bachelor Informatik [PO 22] : Bachelor Informatik [PO 20] : 15/165 					

