

Modulhandbuch Bachelor of Science (B.Sc.)

Informatik [P020]

Stand: Sommersemester 2024

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



Studienplan Bachelor Informatik [PO 20] der Ruhr-Universität Bochum

Nr	Modul	Umfang (LP)	Empfohlenes Semester	Bewertung
Pflichtmodule Mathematik (27 CP)				
1	Mathematik 1	9	1	benotet
2	Mathematik 2	9	2	benotet
3	Mathematik 3	9	3	benotet
Pflichtmodule Informatik (74 CP)				
3	Informatik 1	8	1	benotet
4	Informatik 2	8	2	benotet
5	Informatik 3	8	3	benotet
6	Technische Informatik 1	5	1	benotet
7	Technische Informatik 2	5	2	benotet
8	Technische Informatik 3	5	3	benotet
9	Requirements Engineering	5	1	benotet
10	Verteilte Systeme	5	2	benotet
11	Software Engineering	5	3	benotet
12	Betriebssysteme	5	4	benotet
13	Datenbanksysteme	5	4	benotet
14	Computernetze	5	4	benotet
15	Künstliche Intelligenz	5	4	benotet
Pflichtmodule Englisch				
16	English for Computer Science	3+3	1+2	benotet
17	Presenting and Writing	9	3+4	benotet
Wahlpflichtmodule (25 CP)				
18	Wahlpflichtfach 1	5	5-6	benotet
19	Wahlpflichtfach 2	5	5-6	benotet
20	Wahlpflichtfach 3	5	5-6	benotet
21	Wahlpflichtfach 4	5	5-6	benotet
22	Wahlpflichtfach 5	5	5-6	benotet
Praktische Fächer (22 CP)				
23	Grundlagenpraktikum (Pflicht)	6	4	benotet
24	Praktische Vertiefung (Wahlpflicht): Vertiefungsseminar und Vertiefungspraktikum	6	5-6	benotet
25	Praktische Ausbildung (Pflicht)*	10	5	unbenotet
Fachübergreifende Lehrveranstaltungen				
26	Studium Generale**	5	5-6	benotet
Abschlussarbeit				
27	Bachelorarbeit und Kolloquium	12 + 3	6	benotet

* Hier ist ein Industriepraktikum oder Forschungspraktikum zu absolvieren

** Hier können (nahezu) alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der Universitätsallianz Ruhr gewählt werden.

Angebotene Wahlpflichtmodule

	Lehrveranstaltung	Lehreinheit	Umfang (CP)	Semester	Bewertung
Vertiefungsmodule					
	Algorithmenparadigmen	Informatik	5	WS	benotet
	Artificial Neural Networks	Informatik	6	WS	benotet
	Digitale Forensik	Informatik	5	WS	benotet
	Einführung in die Kryptographie I	Informatik	5	WS	benotet
	Network Planing	ETIT	5	WS	benotet
	Quantum Information and Computation	Informatik	5	WS	benotet
	System Performance Evaluation	Informatik	5	WS	benotet
	Einführung in die Kryptographie II	Informatik	5	SS	benotet
	Functional Programming	Informatik	5	SS	benotet
	Highlights of Theoretical Computer Science	Informatik	10	SS	benotet
	Information Theory	Informatik	5	SS (kein Angebot im SS 24)	benotet
	Model Checking	Informatik	5	SS (kein Angebot im SS 24)	benotet
	Programmanalyse	Informatik	5	SS	benotet
	Public Key Kryptanalyse 1	Informatik	5	SS	benotet
	Systemsicherheit	Informatik	5	SS	benotet
	Web Engineering	BauIng	5	SS	benotet
	Autonomous Vehicles and Artificial Intelligence	Informatik	5	Letztmalig SS 23	benotet
	Game Development	Informatik	6	Letztmalig SS 23	benotet
	Logik in der Informatik	Informatik	5	Letztmalig SS 23	benotet
	Proofs are programs	Informatik	5	Letztmalig SS 23	benotet
	Nebenläufige Programmierung	Informatik	5	Letztmalig SS 23	benotet
	Statistical Learning and Data Mining	Mathematik	5	Letztmalig SS 23	benotet

Angebotene Vertiefungsseminare und Vertiefungspraktika

	Lehrveranstaltung	Lehreinheit	Umfang (CP)	Semester	Bewertung
Vertiefungsseminare					
	Seminar Security Engineering	Informatik	3	WS/SS	benotet
	Seminar Software and Internet Security	Informatik	3	WS/SS	benotet
	Seminar Ressourceneffiziente Systemsoftware	Informatik	3	WS	benotet
	Perlen der theoretischen Informatik (ehemals Grenzen in der theoretischen Informatik)	Informatik	3	WS	benotet
	Seminar Distributed Systems	Informatik	3	WS	benotet
	Algorithms for Decision Making	Informatik	3	WS	benotet
	Seminar Reinforcement Learning	Informatik	3	SS	benotet
	Seminar Quantum Cryptography	Informatik	3	WS	benotet
	Seminar Modern Programming Languages	Informatik	3	WS	benotet
	Seminar Approximationsalgorithmen	Informatik	3	SS (kein Angebot im SS 24)	benotet
	Seminar Quantum Algorithms	Informatik	3	SS (kein Angebot im SS 24)	benotet
	Seminar Perlen der Logik (ehemals Satisfiability)	Informatik	3	SS	benotet
	Seminar From Biological to Artificial Neural Networks	Informatik	3	SS	benotet
	Seminar Safety and Reliability in Artificial Intelligence	Informatik	3	SS	benotet
	Seminar Networked Systems	Informatik	3	SS	benotet
	Seminar Randomisierte Algorithmen	Informatik	3	SS	benotet
	Current topics in microarchitectural security	Informatik	3	SS	benotet
	Seminar Machine Learning Applications	Informatik	3	Letztmalig WS 23/24	benotet
	Seminar on Knowledge Graphs	Informatik	3	Letztmalig WS 22/23	benotet
	Seminar on Current Topics for Systems Security and Privacy	Informatik	3	Letztmalig WS 23/24	benotet

Vertiefungspraktika					
	Android App Evolution	Informatik	3	WS	benotet
	Autonomous Driving Simulation Lab	Informatik	3	WS	benotet
	Praktikum IDE Plugin Development	Informatik	3	WS	benotet
	Praktikum Implementing Post-Quantum Standards and Challenges	Informatik	3	WS	benotet
	Praktikum Systemsoftwaretechnik	Informatik	3	WS	benotet
	Praktikum Rust	Informatik	3	WS	benotet
	Praktikum Worst-Case Performance Evaluation	Informatik	3	WS	benotet
	An Introduction to Python for Data Analysis (ehemals Introduction to Python)	Informatik	3	SS	benotet
	Praktische Kryptanalyse von symmetrischen Chiffren	Informatik	3	SS	benotet
	Python programming and basic machine learning	Informatik	3	SS	benotet

Abkürzungen:

ETIT: Fakultät für Elektrotechnik und Informationstechnik
 Baulng: Fakultät für Bau- und Umweltingenieurwissenschaften

SS: Sommersemester
 WS: Wintersemester
 CP: Creditpoints

MODULHANDBUCH

Übersicht der Module

Informatik - Bachelor (1-Fach, PO 2020)

Pflichtbereich

English for Computer Science
Software Engineering Praktikum
Mathematik 1
Informatik 1
Technische Informatik 1
Requirements Engineering
Mathematik 2
Mathematik 2
Informatik 2
Technische Informatik 2
Verteilte Systeme
Presenting and Writing in English
Mathematik 3
Mathematik 3
Informatik 3
Technische Informatik 3
Software Engineering
Betriebssysteme
Computernetze
Einführung in die künstliche Intelligenz
Datenbanksysteme

Wahlpflichtbereich

Algorithmenparadigmen
Digitale Forensik
Distributed Systems
Einführung in die Kryptographie 1
Einführung in die Kryptographie 2
Functional Programming
Game Development
Highlights of Theoretical Computer Science [B.Sc.]
Information Theory (kein Angebot im SS 24)

Künstliche Neuronale Netze
Model Checking (kein Angebot im SS 24)
Nebenläufige Programmierung (letztmalig SS 23)
Network Planing
Programmanalyse [B.Sc]
Public Key Kryptanalyse 1 [B.Sc]
Quantum Information and Computation
Statistisches Lernen und Data Mining
System Performance Evaluation
Web-Engineering
Vertiefungspraktikum Informatik
Vertiefungsseminar Informatik

Wahlbereich

Studium Generale

Praktische Ausbildung

Praktische Ausbildung PO20

Bachelorarbeit

Abschlussarbeit

Titel des Moduls: English for Computer Science					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester	Turnus Jedes Semester	Dauer 2 Semester
Lehrveranstaltungen <ul style="list-style-type: none"> English for Computer Science 1 (3 CP) English for Computer Science 2 (3 CP) 			Kontaktzeit 60 h	Selbststudium	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		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Frau Melissa Oldfield-Mariano (ZFA) Lehrende: Dozent*innen des Zentrums für Fremdsprachenausbildung (ZFA) der RuhrUniversität Bochum,					
Verwendung des Moduls Dieses Modul wurde exklusiv für diesen Bachelor-Studiengang konzipiert. Daher ist eine Öffnung dieser Veranstaltung für Studierende anderer Studiengänge nicht vorgesehen.					
Vorkenntnisse Englischniveau von ungefähr B2 (GeR). Die bis zum deutschen Abitur erlangten Englischkenntnisse reichen i.d.R. aus. Sollten Zweifel am eigenen Niveau bestehen, wird empfohlen, weitere Lehrveranstaltungen am Zentrum für Fremdsprachenausbildung (ZFA) zur Konsolidierung der Englischkompetenzen zu besuchen.					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls haben die Studierenden das volle Kompetenzniveau B2 in der Fachsprache erreicht, d.h. in diesem Zusammenhang: <ul style="list-style-type: none"> 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 ‚ English for Computer Science I ‘ und auch English for Computer Science II ‘ 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 ‚ English for Computer Science I ‘ 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 ‚**English for Computer Science II**‘ 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 (studienniveauekonform) 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 67 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, Sprachvermittlung, Flipped Classroom, Blended Learning

Prüfungsformen

Zu a) English for Computer Science I:

Semesterbegleitende Leistungsüberprüfung: Portfolio mit einer Sammlung von kurzen Texten (max. 6) (Reading into Writing)

Integratives Prüfungsgespräch mit zwei Prüflingen (Paarprüfung) von 20 Minuten

Zu b) English for Computer Science II:

Semesterbegleitende Leistungsüberprüfung: Portfolio mit einer Sammlung von Hör-Sehtexten (max. 6) (Listening into Speaking) (Audioaufnahmen)

Integratives Prüfungsgespräch mit zwei Prüflingen (Paarprüfung) von 20 Minuten

Voraussetzungen für die Vergabe von Credits

Zu a) und b)

- Anwesenheit von 75 Prozent
- Bestandene Modulabschlussprüfungen sowie erfolgreicher Abschluss aller studienbegleitenden Leistungen

Zu a) English for Computer Science I: Portfolio und Prüfungsgespräch

Zu b) English for Computer Science II: Portfolio und Prüfungsgespräch

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

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

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

Titel des Moduls: Software Engineering Praktikum Software Engineering Lab					
Modul-Nr./Code	Credits 6 CP	Workload 150 h	Semester 4	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					
Vorkenntnisse Programmiererfahrung in einer objektorientierten Programmiersprache (am besten Java), sicherer Umgang mit git					
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) 6/165: B.Sc. Informatik [PO 20]					

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					
Vorkenntnisse					
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]

9/150: B.Sc. IT-Sicherheit [PO 22]

9/149: B.Sc. IT-Sicherheit [PO 20]

Titel des Moduls: Informatik 1 Computer Science 1					
Modul-Nr./Code	Credits 8 CP	Workload 240 h	Semester 1	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Vorlesung und Übung: Informatik 1 (212004)			Kontaktzeit 90 h	Selbststudium 150 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. Informatik [PO 20] B.Sc. IT-Sicherheit [PO 20 + PO 22] B.Sc. Angewandte Informatik [PO 20] B.Sc. Elektrotechnik und Informationstechnik [PO 20 + PO 13]					
Vorkenntnisse keine					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen die Studierenden die wichtigsten Konzepte imperativer und objektorientierter Programmierung • können die Studierenden eigene Programme entwerfen und implementieren • können die Studierenden mit Grundbegriffen der Informatik wie etwa Korrektheit, Laufzeit, Boolesche Algebra, Invarianten und abstrakten Datentypen arbeiten • sind Studierende in der Lage, die einfachen Datenstrukturen (Arrays, Dictionaries) gezielt einzusetzen und kennen Standardalgorithmen darauf, insbesondere zum Sortieren von Arrays 					
Inhalt Zentrales Thema der Veranstaltung ist das Erlernen der Programmierung und der wichtigsten Programmierkonzepte sowie die ersten Grundbegriffe der Informatik: <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, Boolesche Algebra) Die Veranstaltung nutzt die Programmiersprache TScript ("teaching script") für einen möglichst einfachen und motivierenden Einstieg in die Programmierung. Gegen Ende der Vorlesung erfolgt ein Umstieg auf die Programmiersprache Python.					
Lehrformen Vorlesung und Übungen					
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)

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

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 1
Technical Computer Science 1

Modul-Nr./Code	Credits 5 CP	Workload 150 Stunden	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Rechnerarchitektur (141142)			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

Vorkenntnisse

Es werden die Fähigkeit für strukturiertes, algorithmisches Denken sowie das Erfassen von komplexen Abhängigkeiten und Interaktionsmustern vorausgesetzt.

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 zur 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: Requirements Engineering Requirements Engineering					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Requirements Engineering (212001)			Kontaktzeit	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.- Alena Naiakshina Lehrende: Prof. Dr.- Alena Naiakshina					
Verwendung des Moduls B.Sc. Informatik					
Vorkenntnisse Grundkenntnisse der Informatik.					
Lernziele (learning outcomes) Die Veranstaltung vermittelt essentielles Wissen für die systematische Ermittlung, Dokumentation und Verwaltung von Anforderungen in Softwareprojekten. Die Studierenden erlangen ein fundiertes Verständnis von der Bedeutung präziser Anforderungen und deren Auswirkungen auf den gesamten Entwicklungsprozess. Die Vorlesung zielt darauf ab, die Studierenden in die Lage zu versetzen, effiziente Requirements Engineering-Praktiken in ihren zukünftigen Projekten anzuwenden und somit den Erfolg von Softwareentwicklungsprojekten zu fördern.					
Inhalt Effektives Requirements Engineering bildet das fundamentale Rückgrat erfolgreicher Softwareentwicklungsprojekte. Ein solides Verständnis dieses Fachgebiets ist entscheidend, um präzise Anforderungen zu ermitteln und zu verwalten, die den Bedürfnissen und Erwartungen der Stakeholder gerecht werden. Fehlende oder unklare Anforderungen können zu kostspieligen Verzögerungen, unzufriedenen Kunden und potenziell fehlerhafter Software führen. All dies unterstreicht die Bedeutung des Requirements Engineering in der Softwareentwicklung und die Rolle präziser Anforderungen für den Projekterfolg. Studierende lernen verschiedene Anforderungsermittlungsmethoden wie Interviews, Workshops und Umfragen kennen. Zudem werden bewährte Praktiken zur strukturierten Dokumentation und effektiven Verwaltung von Anforderungen vermittelt. Die Validierung von Anforderungen, um ihre Vollständigkeit, Konsistenz und Realisierbarkeit sicherzustellen, ist ein weiterer Schwerpunkt. Darüber hinaus wird die Anforderungsverfolgung und Werkzeuge zur Verwaltung von Änderungen präsentiert. Abschließend wird der Einfluss des Requirements Engineering auf Projektmanagement und Stakeholderkommunikation beleuchtet.					
Lehrformen Vorlesung mit Medienunterstützung und Übungen mit Gruppenarbeit					
Prüfungsformen Schriftliche Modulabschlussprüfung über 90 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 20]					

Titel des Moduls: Mathematik 2 Mathematics 2					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester 2	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 B.Sc. Informatik B.Sc. IT-Sicherheit					
Vorkenntnisse					
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]

9/150: B.Sc. IT-Sicherheit [PO 22]

9/149: B.Sc. IT-Sicherheit [PO 20]

Titel des Moduls: Mathematik 2
Mathematics 2

Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Logik in der Informatik (212013) und Wahrscheinlichkeit für Informatiker (212028)			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr. Thomas Zeume
 Lehrende: Prof. Dr. Thomas Zeume, Prof. Dr. Alex May

Verwendung des Moduls
 B.Sc. Informatik

Vorkenntnisse

Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- kennen die Studierenden lernen wie sich Problemstellungen durch geeignete logische Systeme modellieren lassen,
- beherrschen die Studierenden Syntax und Semantik verschiedener logischer Systeme und können diese nutzen,
- kennen die Studierenden einige klassische logische Kalküle und und können diese durchführen,
- haben die Studierenden ein grundlegendes Verständnis für die Logik-Programmierung entwickelt und können insbesondere einfache Sachverhalte durch Prolog-Programme ausdrücken.
- kennen die Studierende grundlegende Begrifflichkeiten der Wahrscheinlichkeitstheorie wie Zufallsvariablen, sowohl im Diskreten als auch im Kontinuierlichen, und können diese sicher anwenden.
- beherrschen Studierende die Bestimmung von Momenten, wie dem Erwartungswert und der Varianz, im Diskreten mit Hilfe der Technik der z-Transformation.
- erlernen Studierende verschiedene für die Informatik bedeutende Verteilungen wie die Binomial-, Geometrische, Poisson, Uniforme, Exponential- und Normal-Verteilung und können diese sicher anwenden.
- sind Studierende in der Lage, die vorgestellten Techniken mit Hilfe der Computer-Algebra Sage in Python zu implementieren.

Inhalt

Logische Methoden spielen in vielen modernen Anwendungen der Informatik eine wichtige Rolle und der vielfältige Einsatz von Zufallsbits ist grundlegend für die Entwicklung effizienter Algorithmen.

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.

In diesem Modul 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 Folgebeziehungen.

Für viele zentrale Probleme der Informatik sind Lösungen überhaupt nur mit Hilfe von Zufallsbits bekannt. Oft beschleunigen Zufallsbits Algorithmen, oder erlauben eine besonders elegante Analyse der Korrektheit bzw. der Laufzeit. Das Modul verschafft einen Überblick über die vielfältigen Einsatzmöglichkeiten von Zufallsbits im Algorithmen-Design. Die Studierenden erlernen Techniken zum Einsatz von Zufallsbits in Algorithmen (sogenannte probabilistische Algorithmen), sowohl bei der Korrektheits- als auch bei der Laufzeit-Analyse, und implementieren diese in einem Computeralgebra-System.

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 schriftliche Modulabschlussprüfung

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

9/158: B.Sc. Informatik [PO 22]

9/170: B.Sc. Informatik [PO 20]

Titel des Moduls: Informatik 2 Computer Science 2					
Modul-Nr./Code	Credits 8 CP	Workload 240 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Informatik 2 - Algorithmen und Datenstrukturen (211002)			Kontaktzeit 90 h	Selbststudium 150 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Maike Buchin Lehrende: Prof. Maike Buchin					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik B.Sc. IT-Sicherheit/ Informationstechnik					
Vorkenntnisse Inhalte der Module Informatik 1 und Mathematik 1 bzw. Höhere Mathematik 1, insbesondere Programmieren und lineare Algebra					
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: Technische Informatik 2					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Digitaltechnik für ITS und Informatik (211014, ab SoSe 23) Digitaltechnik (141304, bis SoSe 22)			Kontaktzeit 60 h	Selbststudium 90	Gruppengröße Studierende
Unterrichtssprache Deutsch, Vorlesungs- und Übungsfolien auf Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Tim Güneysu Lehrende: Prof. Dr. Tim Güneysu					
Verwendung des Moduls B.Sc. Informatik B.Sc. IT-Sicherheit					
Vorkenntnisse Inhalte des Moduls Mathematik 1 – Grundlagen. Vorausgesetzt wird ein generelles Interesse an technischen Systemen, die Fähigkeit zu strukturieren, algorithmischem Denken sowie die Fähigkeit zum Erfassen von komplexen Abhängigkeiten und Interaktionsmustern.					
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 takt-synchronen 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]

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

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

Titel des Moduls: Verteilte Systeme
distributed systems

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Distributed Systems			Kontaktzeit 45 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende
Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf
Lehrende:

Verwendung des Moduls

B.Sc. Informatik

B.Sc. Angewandte Informatik

Vorkenntnisse

<p>Allgemeines Interesse an technischen Systemen</p>

Lernziele (learning outcomes)

- Die Studierenden sollen eine breite Kenntnis über die auftretenden Herausforderungen beim Entwurf und bei der Anwendung von verteilten Computersystemen erlangen. Nach dem erfolgreichen Abschluss des Moduls 48
- kennen die Studierenden verschiedene Systemmodelle und Architekturen, die zum Entwurf sowie zu der Klassifizierung von verteilten Systemen dienen. Sie können verschiedene Rollen von Teilsystemen differenzieren und sie formal beschreiben
- können die Studierenden vielfältige Herausforderungen beim Aufbau eines verteilten Systems identifizieren und kennen die wichtigsten Standardtechniken zum Umgang mit diesen, inklusive deren Vor- und Nachteile
- können die Studierenden die Funktionsweise eines verteilt implementierten Systems anhand dessen Beschreibung verstehen und die ausgeführte Aufgabe herleiten - können die Studierenden die Fähigkeit eines verteilten Systems zur Erfüllung seiner Aufgabe beurteilen, die Quellen potenzieller Probleme identifizieren und können Verbesserungen sowie deren Integration entwerfen
- sind die Studierenden in der Lage, gegebene Alternativen zur verteilten Implementierung eines Systems für eine bestimmte Aufgabe zu bewerten und begründet in eine Rangfolge zu bringen

Inhalt

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

Lehrformen

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

Prüfungsformen

Schriftliche Modulabschlussprüfung über 120 Minuten

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen

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

5/165: B.Sc. Informatik [PO22]

Titel des Moduls: Presenting and Writing in English

Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester	Turnus Jedes Winter und Sommersemester	Dauer 2 Semester
Lehrveranstaltungen <ul style="list-style-type: none">a) Presenting in English – Vorlesung mit integrierten Übungen (2 SWS, Wintersemester)b) English for Computer Science 2 – Sprachkurs mit integrierten Übungen (2 SWS, Sommersemester)			Kontaktzeit 60 h	Selbststudium 120	Gruppengröße 25 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Frau Melissa Oldfield-Mariano Lehrende: Dozent*innen des Zentrums für Fremdsprachenausbildung (ZFA) der RuhrUniversität Bochum, Nicola Heimann-Bernoussi und Melissa 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.					
Vorkenntnisse					
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 anwendensind die Studierenden in der Lage, ihr Wissen über ein Fachthema bzw. eigene Erkenntnisse sicher zu vermitteln und spontan dazu Fragen zu beantwortensind die Studierenden in der Lage, in einer fachlichen Diskussion persönliche Standpunkte und Meinungen zu äußern und zu erfragen, Stellung zu 70 beziehen, Hypothesen aufzustellen, Argumente und Gegenargumente zu formulieren sowie Vor- und Nachteile aufzuzeigen					
Zu a) 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 aufbereitenSie sind in der Lage, die Vortragsprache (sprachliche Mittel) korrekt einzusetzen und die Präsentation publikumsadäquat vorzutragenSie können auf Fragen zur Präsentation spontan eingehenSie können mit fachrelevanten Vorträgen sicher umgehen.					
Zu b) Writing in English Die Studierenden beherrschen außerdem alle erforderlichen Kompetenzen, um (unkomplizierte) Fachtexte im Englischen schreiben zu können, d.h. in diesem Zusammenhang: <ul style="list-style-type: none">Sie können aus verschiedenen Quellen Informationen für ihren Text zusammentragenSie können diese Informationen fachkonform und den Konventionen der englischen Sprache angemessen aufbereitenSie 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 fachkonformen 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 integrativer Bestandteil beider Lehrveranstaltungen

Lehrformen

Gruppenarbeiten, Projektarbeit, Sprachvermittlung, Flipped Classroom, Blended Learning

Prüfungsformen

Zu a) Presenting in English: Präsentationen in verschiedenen Formaten

Zu b) Writing in English: Schriftliche Aufgaben unterschiedlicher Formate; Portfolio mit der Sammlung von Texten

Voraussetzungen für die Vergabe von Credits

Zu a) und b)

- Anwesenheit von 75 Prozent
- Erfolgreicher Abschluss aller studienbegleitenden Leistungen

Zu a) Presenting in English Erfolgreiches Erstellen, Bearbeiten und Durchführen der Präsentationen

Zu b) Writing in English Erfolgreiches Erarbeiten aller schriftlichen Aufgaben und Abgabe des vollständigen Portfolios

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 3	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Logik in der Informatik (212013) und Probabilistische Algorithmen (212028)			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße 100 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		

Modulbeauftragte/r und hauptamtlich Lehrende
 Modulbeauftragte/r: Prof. Dr. Thomas Zeume
 Lehrende: Prof. Dr. Thomas Zeume, Prof. Dr. Alex May

Verwendung des Moduls

B.Sc. Informatik

Vorkenntnisse

Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- kennen die Studierenden lernen wie sich Problemstellungen durch geeignete logische Systeme modellieren lassen,
- beherrschen die Studierenden Syntax und Semantik verschiedener logischer Systeme und können diese nutzen,
- kennen die Studierenden einige klassische logische Kalküle und und können diese durchführen,
- haben die Studierenden ein grundlegendes Verständnis für die Logik-Programmierung entwickelt und können insbesondere einfache Sachverhalte durch Prolog-Programme ausdrücken.
- kennen die Studierende grundlegende Begrifflichkeiten wie diskrete Zufallsvariablen, Momente und Verteilungen,
- beherrschen Studierende die Abschätzung von Laufzeiten von Algorithmen mit Hilfe von Chernoff-Schranken,
- können sie zufällige Walks auf Graphen modellieren und mit Hilfe von Markovketten analysieren,
- erlernen Studierende den Entropie-Begriff und können Daten optimal verlustfrei komprimieren,
- beherrschen Studenten die Analyse und Anwendung universeller Hashfunktionen,
- sind Studierende in der Lage, die vorgestellten Techniken mit Hilfe der Computer-Algebra Sage zu implementieren.

Inhalt

Logische Methoden spielen in vielen modernen Anwendungen der Informatik eine wichtige Rolle und der vielfältige Einsatz von Zufallsbits ist grundlegend für die Entwicklung effizienter Algorithmen.

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.

In diesem Modul 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.

Für viele zentrale Probleme der Informatik sind Lösungen überhaupt nur mit Hilfe von Zufallsbits bekannt. Oft beschleunigen Zufallsbits Algorithmen, oder erlauben eine besonders elegante Analyse der Korrektheit bzw. der Laufzeit. Das Modul verschafft einen Überblick über die vielfältigen Einsatzmöglichkeiten von Zufallsbits im Algorithmen-Design. Die Studierenden erlernen Techniken zum Einsatz von Zufallsbits in Algorithmen (sogenannte probabilistische Algorithmen), sowohl bei der Korrektheits- als auch bei der Laufzeit-Analyse, und implementieren diese in einem Computeralgebra-System.

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 schriftliche Modulabschlussprüfung

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

9/158: B.Sc. Informatik [PO 22]

9/170: B.Sc. Informatik [PO 20]

Titel des Moduls: Mathematik 3 Mathematics 3					
Modul-Nr./Code	Credits 9 CP	Workload 270 h	Semester 3	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen N.N.			Kontaktzeit 90 h	Selbststudium 180 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr. Christof Beierle Lehrende: Dr. Christof Beierle					
Verwendung des Moduls B.Sc. Informatik					
Vorkenntnisse					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende grundlegende Begriffe und Techniken der reellen Analysis in einer Variablen, sowie Grundzüge der mehrdimensionalen Analysis • verstehen die Studierenden insbesondere den Konvergenzbegriff, der Begriff der Differenzierbarkeit, sowie den Begriff des Riemann-Integrals und Anwendungen dieser • können Studierende die erlernten Techniken selbstständig anwenden. 					
Inhalt Dieses Modul vermittelt die Grundlagen in der reellen Analysis in einer Variablen, sowie eine Einführung in die mehrdimensionale Analysis. Die Vorlesung basiert auf ausgewählten Kapiteln des Buches "Analysis für Informatiker" von M. Oberguggenberger und A. Ostermann. Behandelte Themen sind: <ul style="list-style-type: none"> • Einführung in die reellen Zahlen und reellwertigen Funktionen • Winkelfunktionen • Komplexe Zahlen • Folgen und Reihen, Konvergenzbegriff, Vollständigkeit der reellen Zahlen • Grenzwerte und Stetigkeit von Funktionen • Differenzierbarkeit, Ableitungen, Ableitungsregeln • Extremwerte • Newtonverfahren zur Berechnung von Nullstellen differenzierbarer Funktionen • Stammfunktionen • Riemann-Integrale • Hauptsätze der Differential- und Integralrechnung • Skalarwertige Funktionen in zwei Veränderlichen, partielle und totale Ableitungen, Richtungsableitung und Gradient • Vektorwertige Funktionen in zwei Veränderlichen, Jacobimatrix, Newtonverfahren in zwei Variablen 					
Lehrformen Vorlesung mit Übung					
Prüfungsformen Schriftliche Modulabschlussprüfung über 120 Minuten					

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

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

9/158: B.Sc. Informatik [PO 22]

9/170: B.Sc. Informatik [PO 20]

Titel des Moduls: Informatik 3 Computer Science 3					
Modul-Nr./Code	Credits 8 CP	Workload 240 h	Semester 3	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Informatik 3 - Theoretische Informatik (212002)			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					
Vorkenntnisse					
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 (180 Minuten)					

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

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

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

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

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

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

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

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

Titel des Moduls: Technische Informatik 3

Modul-Nr./Code	Credits 5 CP	Workload 150 Stunden	Semester 3	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Technische Informatik 3 - Hardwareprogrammierung (212003)			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 B.Sc. Informatik B.Sc. Angewandte Informatik					
Vorkenntnisse Inhalte der Module Informatik 1 – Programmierung, Informatik 2 – Algorithmen und Datenstrukturen, Technische Informatik 1 – Rechnerarchitektur und Technische Informatik 2 – Digitaltechnik					
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 150 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]

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

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

Titel des Moduls: Software Engineering Software Engineering					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 3	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen 212000: Software Engineering			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 350 Studierende
Unterrichtssprache			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
Verwendung des Moduls					
Vorkenntnisse					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen					
Prüfungsformen					
Voraussetzungen für die Vergabe von Credits					
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

Vorkenntnisse

<p>Grundkenntnisse der Informatik (Inhalte der Module Informatik 1 – Programmierung und Technische Informatik 1 – Rechnerarchitektur)</p>

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]

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

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

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

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

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

Titel des Moduls: Computernetze Computer Networks					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester 2	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Computernetze (211006)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 400 Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Dr.-Ing. Christian Mainka Lehrende: Dr.-Ing. Christian Mainka					
Verwendung des Moduls B.Sc. Informatik B.Sc. Angewandte Informatik B.Sc. IT-Sicherheit / Informationstechnik					
Vorkenntnisse					
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 150 h	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. Sen Cheng Lehrende: Prof. Dr. Laurenz Wiskott, Prof. Dr. Tobias Glasmachers, Prof. Dr. Sen Cheng, Prof. Dr. Gregor Schöner, Prof. Dr. Nils Jansen Prof. Dr. Setareh Maghsudi Prof. Dr. Christian Straßer					
Verwendung des Moduls B.Sc. Informatik (Pflichtmodul) B.Sc. Angewandte Informatik (Pflichtmodul) B.Sc. IT-Sicherheit/Informationstechnik (Wahlpflichtmodul)					
Vorkenntnisse					
Lernziele (learning outcomes)					
Inhalt					
Lehrformen					
Prüfungsformen Written module final exam (120 minutes)					
Voraussetzungen für die Vergabe von Credits					
Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS) 5/158: B.Sc. Informatik [PO 22] 5/170: B.Sc. Informatik [PO 20] 5/168: B.Sc. Angewandte Informatik [PO 22] 5/170: B.Sc. Angewandte Informatik [PO 20]					

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

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

Titel des Moduls: Datenbanksysteme Database Systems					
Modul-Nr./Code	Credits 5 CP	Workload 210 h	Semester 4	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Datenbanksysteme (211008)			Kontaktzeit 60 h	Selbststudium 150 h	Gruppengröße 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]					
Vorkenntnisse Grundkenntnisse der Informatik (Inhalte der Module Informatik 1 – Programmierung, Technische Informatik 1 – Rechnerarchitektur)					
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: Algorithmenparadigmen					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	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					
Vorkenntnisse Inhalte aus dem Modulen (Höhere) Mathematik 1 und 2 sowie aus den Modulen Infomatik 1-3					
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) oder mündliche Prüfung					
Voraussetzungen für die Vergabe von Credits Bestandene schriftliche oder mündliche 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: Digitale Forensik					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Digitale Forensik (211017)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: Dr. Christofer Fein					
Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik B.Sc. Informatik M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse <p>Erfahrung in systemnaher Programmierung sowie der Programmiersprache C sind hilfreich für das Verständnis der vermittelten Themen.</p>					
Lernziele (learning outcomes) Die Studierenden beherrschen verschiedene Konzepte, Techniken und Tools aus dem Themengebiet der digitalen Forensik. Sie kennen die relevanten Konzepte und haben ein Überblick zum Forensischen Prozess. Es ist grundlegendes Verständnis von verschiedenen Methoden zur Sammlung, Analyse und Aufbereitung digitaler Spuren in IT-Systemen vorhanden. Die Studierenden können eigenständig neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Sie können mit diesem Verständnis mit ihren Kollegen über Probleme der Computerforensik diskutieren und auftretende Probleme im Gespräch korrekt klassifizieren.					
Inhalt Digitale Forensik befasst sich mit der Sammlung, Analyse und Aufbereitung digitaler Spuren in IT-Systemen. Im Rahmen der Vorlesung werden diese drei Themenbereiche vorgestellt und jeweils erläutert, mit welchen Verfahren und Ansätzen man diese Aufgaben erreichen kann. Ein Schwerpunkt der Vorlesung liegt auf dem Bereich der Analyse von Dateisystemen. Dazu werden verschiedene Arten von Dateisystemen detailliert vorgestellt und diskutiert, wie relevante Daten erfasst, analysiert und aufbereitet werden können. Darüber hinaus werden weitere Themen aus dem Bereich der digitalen Forensik behandelt, z.B. die Analyse von Smartphones und SQLite-Datenbanken. Ein integraler Teil der Veranstaltung sind die Übungen, die den Stoff mit praktischen Beispielen verdeutlichen und vertiefen.					
Lehrformen Vorlesung mit Übung als Blockveranstaltung					
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) 5/150: B.Sc. IT-Sicherheit/ Informationstechnik [PO 22]					

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

5/170: B.Sc. Informatik [PO 22]

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

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: Distributed Systems

Distributed systems

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

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf
Lehrende: Prof. Dr.-Ing. Steffen Bondorf

Verwendung des Moduls

B.Sc. Angewandte Informatik

B.Sc. Informatik [unter dem deutschen Modultitel]

Vorkenntnisse

<p>Allgemeines Interesse an technischen Systemen</p>

Lernziele (learning outcomes)

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

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

Inhalt

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

Lehrformen

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

Prüfungsformen

Schriftliche Modulabschlussprüfung über 120 Minuten

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen

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

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

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

Titel des Moduls: Einführung in die Kryptographie 1

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	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 M.Sc. IT-Sicherheit/ Netze und Systeme					
Vorkenntnisse					
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 in der Praxis eingesetzten Lösungen. Sie sind in der Lage, argumentativ eine bestimmte Lösung zu verteidigen. Die Vorlesungen werden auch als Videos in Deutsch und Englisch angeboten. Die Studierenden können daher durch das zweisprachige eLearning-Angebot Sprachkompetenzen in der Wissenschaftssprache Englisch erwerben.					
Inhalt Das Modul bietet einen allgemeinen Einstieg in die Funktionsweise moderner Kryptographie und ihrer Bedeutung für die IT-Sicherheit. Es werden grundlegende Begriffe und mathematisch/technische Verfahren der Kryptographie erläutert. Praktisch relevante symmetrische und asymmetrische Verfahren und Algorithmen werden vorgestellt und an praxisrelevanten Beispielen erläutert. Die Vorlesung lässt sich in zwei Teile gliedern: Die Grundlagen der symmetrischen Kryptographie einschließlich der Beschreibung einiger historischer Verschlüsselungsverfahren (Caesar Chiffre, Affine Chiffre), aktueller symmetrischer Verfahren (AES, 3-DES) und grundlegender Konzepte wie dem One-Time-Pad und Stromchiffren werden im ersten Teil behandelt. Benötigte mathematische Grundlagen, insbesondere modulares Rechnen und endliche Körper, werden ebenfalls aus					

Anwendersicht eingeführt.

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

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

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Klausurarbeit (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Erfolgreiches Bestehen der Modulklausur.

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

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

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

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

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

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

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

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

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

Titel des Moduls: Einführung in die Kryptographie 2

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	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
M.Sc. IT-Sicherheit/ Netze und Systeme

Vorkenntnisse

Inhalte der Vorlesung "Einführung in die Kryptographie 1"

Lernziele (learning outcomes)

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

Inhalt

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

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

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

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

Lehrformen

Vorlesung mit Übungen

Prüfungsformen

Klausurarbeit (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung.

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

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

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

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

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

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

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

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

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

Titel des Moduls: Functional Programming
Functional Programming

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

Modulbeauftragte/r und hauptamtlich Lehrende

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

Verwendung des Moduls

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

Vorkenntnisse

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

Lernziele (learning outcomes)

Nach erfolgreichem Abschluss dieses Kurses können die Studierenden:

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

Inhalt

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

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

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

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestehen der Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen.

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

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

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

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

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

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

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

Titel des Moduls: Game Development Game Development					
Modul-Nr./Code	Credits 6 CP	Workload 180 h	Semester siehe Prüfungsordnung	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					
Vorkenntnisse Grundlegende Programmierkenntnisse; Zugang zu einem Computer mit Internet, auf dem die Unity-Engine läuft (https://unity3d.com/de/get-unity/download)					
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: Highlights of Theoretical Computer Science [B.Sc.]
Highlights of Theoretical Computer Science [B.Sc.]

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

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

Lehrformen

Vorlesung mit Übung

Prüfungsformen

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

Voraussetzungen für die Vergabe von Credits

Bestandene Modulabschlussprüfung

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

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

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

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

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

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

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

Titel des Moduls: Information Theory (kein Angebot im SS 24)

Information Theory (no offer in SS 24)

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Information Theory (211007)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße 30 Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen Keine		

Modulbeauftragte/r und hauptamtlich Lehrende

Modulbeauftragte/r: Prof. Dr. Michael Walter

Lehrende: Prof. Dr. Michael Walter

Verwendung des Moduls

B.Sc. Informatik (bis SS 23)

B.Sc. IT-Sicherheit

M.Sc. Informatik

M.Sc. IT-Sicherheit/ Informationstechnik

M.Sc. IT-Sicherheit/ Netze und Systeme (bis SS 23)

M.Sc. Angewandte Informatik

Vorkenntnisse

Vertrautheit mit der diskreten Wahrscheinlichkeitsrechnung (wir werden Sie kurz an die wichtigsten Fakten erinnern). Einige Erfahrung mit präzisen mathematischen Aussagen und strengen Beweisen (da wir viele davon im Kurs sehen werden). Ein Teil der Hausaufgaben wird die Programmierung in Python erfordern.

Lernziele (learning outcomes)

Sie werden grundlegende Konzepte, Algorithmen und Ergebnisse der Informationstheorie kennenlernen.

Nach erfolgreichem Abschluss dieses Kurses kennen Sie das mathematische Modell der Informationstheorie, wissen, wie man Algorithmen für eine Vielzahl von Informationsverarbeitungsaufgaben entwirft und analysiert, und wie man sie in Python implementiert. Sie haben sich selbstständig in ein Thema der Informationstheorie eingelesen und dieses vor Ihren Kommilitonen präsentiert. Sie werden auf einen weiterführenden Kurs oder ein Forschungs- oder Abschlussprojekt in diesem Bereich vorbereitet. Eine genaue Auflistung der Lernziele finden Sie auf der Homepage des Kurses.

Inhalt

Dieser Kurs gibt eine Einführung in die Informationstheorie - die mathematische Theorie der Information. Seit ihren Anfängen hat die Informationstheorie einen tiefgreifenden Einfluss auf die Gesellschaft gehabt. Sie bildet die Grundlage für wichtige technologische Entwicklungen, von zuverlässigen Speichern bis hin zu Mobilfunkstandards, und ihr vielseitiges mathematisches Instrumentarium findet Anwendung in der Informatik, dem maschinellen Lernen, der Physik, der Elektrotechnik, der Mathematik und vielen anderen Disziplinen.

Ausgehend von der Wahrscheinlichkeitstheorie werden wir erörtern, wie man Informationsquellen und Kommunikationskanäle mathematisch modelliert, wie man Informationen optimal komprimiert und wie man fehlerkorrigierende Codes entwirft, die uns eine zuverlässige Kommunikation über verrauschte Kommunikationskanäle ermöglichen. Wir werden auch sehen, wie die in der Informationstheorie verwendeten

Techniken allgemeiner angewendet werden können, um Vorhersagen aus verrauschten Daten zu treffen.

Vorläufiger Lehrplan:

- Begrüßung, Einführung in die Informationstheorie
- Auffrischung der Wahrscheinlichkeitstheorie
- Numerische Zufallsvariablen, Konvexität und Konkavität, Entropie
- Symbol-Codes: Verlustfreie Komprimierung, Huffman-Algorithmus
- Block-Codes: Shannons Quellencodierungstheorem, sein Beweis und Variationen
- Strom-Codes: Lempel-Ziv-Algorithmus
- Strom-Codes: Arithmetische Kodierung
- Gemeinsame Entropien & Kommunikation über verrauschte Kanäle
- Shannons Theorem der verrauschten Kodierung
- Beweis des Theorems der verrauschten Kodierung (Noisy Coding Theorem)
- Beweis der Umkehrung, Shannons Theorie und Praxis
- Reed-Solomon-Codes
- Nachrichtenübermittlung für Dekodierung und Inferenz, Ausblick
- Studentische Präsentationen

Weitere Informationen finden Sie auf der Kurs-Homepage https://qi.rub.de/it_ss23.

Lehrformen

Vorlesung mit Übung

Prüfungsformen

Schriftliche (180 Minuten) oder mündliche (30 Minuten) Modulabschlussprüfung, abhängig von der Teilnehmerzahl. Wird zum Kursbeginn bekanntgegeben.

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]

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

5/105: M.Sc. Angewandte Informatik

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

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

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

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

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

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

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

Titel des Moduls: Model Checking (kein Angebot im SS 24)
Model Checking (no offer in SS 24)

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Model Checking (211000)			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

Vorkenntnisse

<p>Grundlagenvorlesungen Mathematik,</p>
 <p>Einführung in die Theoretische Informatik (ggf. kann das nötige Wissen auch nachgeholt werden),</p>
 <p>Hilfreich: Logik in der Informatik, Datenstrukturen und elementare Programmierkenntnisse</p>

Lernziele (learning outcomes)

Die Studierenden lernen wie sich verteilte Systeme durch Transitionssysteme modellieren und Eigenschaften in logischen Spezifikationssprachen 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 Spezifikationssprachen. Die Spezifikationssprachen 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 (letztmalig SS 23)

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	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 Master IT-Sicherheit/ Informationstechnik					
Vorkenntnisse Beherrschung einer Objektorientierten Programmiersprache (idealerweise Java)					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• 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]

5/91 Master IT-Sicherheit/ Informationstechnik

Titel des Moduls: Network Planing

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	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

B.Sc. Informatik

Vorkenntnisse

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: Programmanalyse [B.Sc]					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Programmanalyse (211015)			Kontaktzeit 60 h	Selbststudium 90 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Kevin Borgolte Lehrende: Prof. Kevin Borgolte					
Verwendung des Moduls B.Sc. IT-Sicherheit/ Informationstechnik B.Sc. Informatik					
Vorkenntnisse keine					
Lernziele (learning outcomes) Die Studierenden kennen verschiedene Konzepte, Techniken und Tools aus dem Bereich der Programmanalyse. Dies beinhaltet den Überblick über verschiedene Konzepte aus dem Bereich Reverse Engineering sowie Binäranalyse. Die Studierenden haben grundlegendes Verständnis von sowohl statischen als auch dynamischen Methoden zur Analyse eines gegebenen Programms. Sie sind in der Lage, verschiedene Aspekte der Programmanalyse zu beschreiben und auf neue Problemstellungen anzuwenden.					
Inhalt In der Vorlesung werden unter anderem die folgenden Themen und Techniken aus dem Bereich der Programmanalyse behandelt: <ul style="list-style-type: none"> • Statische und dynamische Analyse von Programmen • Analyse von Kontroll- und Datenfluss • Symbolische Ausführung • Taint Tracking • Binary Instrumentation • Program Slicing • Überblick zu existierenden Analysetools Daneben wird im ersten Teil der Vorlesung eine Einführung in x86/x64 Assembler gegeben sowie die grundlegenden Techniken aus dem Themenbereich Reverse Engineering vorgestellt. Begleitet wird die Vorlesung von Übungen, in denen die vorgestellten Konzepte und Techniken praktisch eingeübt werden.					
Lehrformen Vorlesung mit Übung					
Prüfungsformen mündliche oder schriftliche Modulabschlussprüfung (wird zu Beginn des Semester bekanntgegeben), Anmeldung: FlexNow					
Voraussetzungen für die Vergabe von Credits Bestandene Modulabschlussprüfung					

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

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

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

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

5/170: B.Sc. Informatik [PO 22]

Titel des Moduls: Public Key Kryptanalyse 1 [B.Sc]
Public Key Cryptanalysis 1 [B.Sc]

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Sommersemester	Dauer Semester
Lehrveranstaltungen Public Key Kryptanalyse 1 (211055)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Deutsch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Alex May Lehrende: Prof. Alex May					
Verwendung des Moduls					
Vorkenntnisse Vorausgesetzt werden elementare Kenntnisse der Lineare Algebra (Mathematik 1 & Informatiker) und ein Interesse an algorithmischen Techniken und Kryptographie, in Theorie und Praxis (umgesetzt mit Hilfe des Computeralgebra-Systems Sage).					
Lernziele (learning outcomes) Die Studierenden sollen breite Kenntnisse zu algorithmischen Techniken der asymmetrischen Kryptanalyse, insbesondere für codierungsbasierte Kryptographie, erlangen. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen die Studierenden grundlegende Schlüsselfindungs-Algorithmen wie Brute-Force und Meet-in-the-Middle und können diese auf neue kryptographische Systeme anwenden, • beherrschen sie die Grundlagen linearer Codes und ihrer Dualcodes, insbesondere als kryptographische Anwendung das McEliece-Kryptosystem, • kennen Studierende Time-Memory Techniken wie Pollard Rho und Parallel Collision Search, und können sie auf neue Probleme anwenden, • haben Studierende einen Überblick über alle aktuellen Dekodieralgorithmen im Bereich des Information Set Decoding, die für die Sicherheits-Evaluierung moderner kodierungsbasierter Kryptosysteme relevant sind, • sind Studierende in der Lage, Techniken der Kryptanalyse mit Hilfe der Computer-Algebra Sage zu implementieren. 					
Inhalt Kryptanalyse dient dazu, kryptographische Systeme derart zu instantiiieren, dass sie einerseits ein vordefiniertes Sicherheitsniveau bieten, andererseits aber möglichst performant sind. Die Kryptanalyse bietet dazu einen ganzen Werkzeugkoffer an algorithmischen Techniken, um die Evaluation neuer kryptographischer Systeme zu realisieren. Dies beinhaltet sowohl klassische Algorithmen als auch Algorithmen für Quantenrechner, damit die verwendete Kryptographie selbst in einer Ära von Quantenrechnern sicher bleiben.					
Lehrformen Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner mit der Computer-Algebra Sage werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
Prüfungsformen Schriftliche Modulabschlussprüfung über 120 Minuten					
Voraussetzungen für die Vergabe von Credits					

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

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

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

Titel des Moduls: Quantum Information and Computation
Quantum Information and Computation

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen Quantum Information and Computation (212011)			Kontaktzeit 4 SWS (60 h)	Selbststudium 90 h	Gruppengröße 40 Studierende
Unterrichtssprache Deutsch oder Englisch (depends on audience)			Teilnahmevoraussetzungen Keine		

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
 M.Sc. IT-Sicherheit/ Informationstechnik (auf Antrag)
 M.Sc. IT-Sicherheit/ Netze und Systeme (auf Antrag)
 M.Sc. Computer Science

Vorkenntnisse

<p>Familiarity with linear algebra (in finite dimensions) and
probability (with finitely many outcomes) at the level of a first Bachelors course; we will briefly remind you of the more difficult bits in class. In addition, some mathematical maturity, since we will discuss precise mathematical statements and rigorous proofs. No background in physics is required.</p>

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:

- 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

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.

Lehrformen

Lecture with Exercise

Prüfungsformen

Final written module exam (180 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

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

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

5/97: M.Sc. Computer Science

Titel des Moduls: Statistisches Lernen und Data Mining
Statistical learning and data mining

Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung / see examination regulations	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Statistisches Lernen und Data Mining (150331)			Kontaktzeit 60 h	Selbststudium 90	Gruppengröße 20 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. Computer Science					
Vorkenntnisse					
Lernziele (learning outcomes) Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • 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. 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 (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] 5/97: M.Sc. Computer Science					

Titel des Moduls: System Performance Evaluation System Performance Evaluation					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester siehe Prüfungsordnung	Turnus Wintersemester	Dauer 1 Semester
Lehrveranstaltungen System Performance Evaluation (212033)			Kontaktzeit 45 h	Selbststudium 105 h	Gruppengröße Studierende
Unterrichtssprache Englisch			Teilnahmevoraussetzungen keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf Lehrende: Prof. Dr.-Ing. Steffen Bondorf					
Verwendung des Moduls B.Sc. Informatik M.Sc. Computer Science					
Vorkenntnisse <p>Empfohlen: Inhalte der Module Technische Informatik 1, Computernetze und System Performance Evaluation</p>					
Lernziele (learning outcomes) Die Studierenden sollen eine breite Kenntnis über folgende Themenfelder erlangen: <ul style="list-style-type: none"> • Systemmodellierung und -leistungsanalyse • Entwurf von Experimenten • Messungen • Simulation 					
Inhalt Die Veranstaltung „System Performance Evaluation“ vermittelt grundlegende Techniken für Modellierung von Computersystemen und der Quantifizierung ihrer Leistung. Im Vordergrund stehen der Entwurf von Experimenten, Simulation und Warteschlangen-theorie.					
Lehrformen Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner können weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
Prüfungsformen schriftliche (90 Minuten) oder mündliche Modulabschlussprüfung					
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] 5/170: B.Sc. Informatik [PO 20] 5/97: M.Sc. Computer Science					

Titel des Moduls: Vertiefungspraktikum Informatik					
Modul-Nr./Code	Credits 3 CP	Workload 90 h	Semester 5	Turnus Wintersemester oder Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Aktuelle Praktika sind hervorgehoben: <ul style="list-style-type: none">• Praktikum Systemsoftware-Technik (212405)• Praktikum Implementing Post Quantum Standards and Challenges (212416)• Praktikum IDE Plugin Development (212418)• Android App Evolution (212419)• Autonomous Driving Simulation Lab (212421)• An Introduction to Python for Data Analysis (211421)• Praktische Kryptanalyse von symmetrischen Chiffren (211401)• Python programming and basic machine learning (211434)• Praktikum Rust (212407)			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 Informatik Lehrende: siehe jeweiligen Praktikumseintrag					
Verwendung des Moduls B.Sc. Informatik					
Vorkenntnisse Programmieren, ggf. weitere Vorkenntnisse abhängig vom Praktikum					
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• An Introduction to Python for Data Analysis• Praktische Kryptanalyse von symmetrischen Chiffren• Python programming and basic machine learning• Praktikum Rust• Autonomous Driving Simulation Lab					

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 Die aktuellen Seminare sind hervorgehoben:			Kontaktzeit 30 h	Selbststudium 60 h	Gruppengröße Studierende
<ul style="list-style-type: none"> • Seminar Perlen der Logik (ehemals 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 bzw. 211135) • Seminar Ressourceneffiziente Systemsoftware (212111) • Seminar Modern Programming Languages (212115) • Seminar Quantum Algorithms (211119) • Seminar Safety and Reliability Artificial Intelligence (211138) • Seminar Randomisierte Algorithmen (211139) • Seminar Networked Systems (211137) • Seminar: Current topics in microarchitectural security (211134) • Seminar Software and Internet Security (212125) • Seminar Implementation Security (212126) ausgelaufen • Seminar zur symmetrischen Kryptographie (212118) ausgelaufen • Seminar Knowledge Graphs (212113) ausgelaufen • Machine Learning Applications (212108) ausgelaufen 					
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) 					

Vorkenntnisse

Die Vertiefungsseminare beziehen sich in der Regel auf Inhalte aus bestimmten Pflicht- oder Vertiefungsmodulen, die im Vorfeld absolviert worden sein sollten.

Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- 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)

3/158: B.Sc. Informatik [PO 22]

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

Titel des Moduls: Web-Engineering Web-Engineering					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester Siehe Prüfungsordnung	Turnus Sommersemester	Dauer 1 Semester
Lehrveranstaltungen Web-Engineering (128968 + 128969)			Kontaktzeit 60 h	Selbststudium 90 h	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					
Vorkenntnisse Programmieren, Architektur von Web-basierten Systemen					
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 Moduls werden den Studierenden aktuelle Techniken und Kenntnisse im Bereich der Web-Entwicklung aufgezeigt. Thematisch wird der Bereich der server- und clientseitigen Entwicklung abgedeckt. JavaScript stellt dabei eine zentrale Rolle dar. Folgende Lehrinhalte werden behandelt: <ul style="list-style-type: none"> • 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 Synchrone Onlinevorlesung, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend					

bereitgestellten Materialien und Aufgaben.

Prüfungsformen

Schriftliche Modulabschlussprüfung (120 Minuten)

Voraussetzungen für die Vergabe von Credits

Bestandene schriftliche Modulabschlussprüfung

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

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

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

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

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

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

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

Titel des Moduls: Studium Generale					
Modul-Nr./Code	Credits 5 CP	Workload 150 h	Semester	Turnus Jedes Semester	Dauer Semester
Lehrveranstaltungen Nichttechnische Wahlmodule Die Lehrveranstaltungen sind aus dem nichttechnischen Bereich frei wählbar.			Kontaktzeit 60 h	Selbststudium	Gruppengröße Studierende
Unterrichtssprache Abhängig von der jeweiligen Veranstaltung			Teilnahmevoraussetzungen Keine		
Modulbeauftragte/r und hauptamtlich Lehrende Modulbeauftragte/r: Modulbeauftragte/r: Studiendekan*in Informatik Hauptamtlich Lehrende/r: Siehe den jeweiligen Eintrag im Vorlesungsverzeichnis Lehrende:					
Verwendung des Moduls Unterschiedliche Studiengänge, abhängig vom gewählten Modul					
Vorkenntnisse					
Lernziele (learning outcomes) Innerhalb des Moduls setzen die Studierenden entsprechend ihrer Interessen verschiedene Schwerpunkte. Dafür steht ihnen das breite Angebot der gesamten Universität zur Verfügung. Sie beherrschen entsprechend ihrer Auswahl verschiedene Schlüssel-qualifikationen. Die vermittelte Kernkompetenz ist fächerübergreifendes Denken und Arbeiten.					
Inhalt Die nichttechnischen Wahlfächer erweitern die Soft Skills der Studierenden. Beispielsweise werden die Kompetenzen in einer Fremdsprache verbessert oder es wird eine neue Fremdsprache erlernt, wird in die Grundlagen der 135 Rechtswissenschaften eingeführt oder es werden Grundkenntnisse der Betriebswirtschaft vermittelt. Die Studierenden haben die Möglichkeit, eine Auswahl entsprechend der eigenen Interessen zu treffen.					
Lehrformen Abhängig von der gewählten Lehrveranstaltung, siehe die jeweilige Beschreibung im Vorlesungsverzeichnis					
Prüfungsformen Siehe den jeweiligen Eintrag im Vorlesungsverzeichnis					
Voraussetzungen für die Vergabe von Credits					
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: Praktische Ausbildung PO20					
Modul-Nr./Code	Credits 10 CP	Workload 300 h	Semester 5	Turnus Jedes Winter und Sommersemester	Dauer 1 Semester
Lehrveranstaltungen <ul style="list-style-type: none"> • Industriepraktikum • Software-Projekt • Forschungsprojekt 			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					
Vorkenntnisse					
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-Entwicklungscompetenz 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] 					
Vorkenntnisse					
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)

- Bachelor Informatik [PO 22] :
- Bachelor Informatik [PO 20] : 15/165