

RUHR-UNIVERSITÄT BOCHUM

FACULTY OF COMPUTER SCIENCE

RUB

# Modulhandbuch

für den Bachelorstudiengang

# INFORMATIK

STAND 30.05.2023

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



Faculty of  
Computer  
Science

[www.informatik.rub.de](http://www.informatik.rub.de)



## **Wo finde ich Hilfe während des Studium?**

### **Wichtige Informationsquellen rund um das Studium der Informatik:**

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

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

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

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

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

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

### **Weitere wichtige Kontaktadressen auf dem Campus sind:**

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

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

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

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

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

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

## Studiengangsziele:

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

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

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

## Modularisierungskonzept:

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

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

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

## Prüfungsformen:

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

## Studienverlaufsplan:

Nr	Modul	Umfang (LP)	Empfohlenes Semester	Bewertung
<b>Pflichtbereich</b>				
1	Mathematik 1	9	1	benotet
2	Informatik 1	8	1	benotet
3	Technische Informatik 1	5	1	benotet
4	Requirements Engineering	5	1	benotet
4	English for Computer Science 1 und II	3+3	1+2	benotet
5	Mathematik 2	9	2	benotet
6	Informatik 2	8	2	benotet
7	Technische Informatik 2	5	2	benotet
8	Verteilte Systeme	5	2	benotet
9	Presenting and Writing in English	3+3	3+4	benotet
10	Mathematik 3	9	3	benotet
11	Informatik 3	8	3	benotet
12	Software Engineering	5	3	benotet
13	Technische Informatik 3	5	3	benotet
14	Einführung in die künstliche Intelligenz	5	4	benotet
15	Datenbanksysteme	5	4	benotet
16	Computernetze	5	4	benotet
17	Betriebssysteme	5	4	benotet
18	Software Engineering Praktikum	6	4	benotet
<b>Wahlpflichtbereich</b>				
19	Vertiefungspraktikum	3	5	benotet
20	Vertiefungsseminar	3	5	benotet
21	Vertiefungsmodule	25	5-6	benotet
<b>Wahlbereich</b>				
22	Studium Generale	5	1-6	benotet
<b>Praktische Ausbildung</b>				
23	Praktische Ausbildung	10	5-6	unbenotet
<b>Abschlussarbeit</b>				
24	Bachelorarbeit und Kolloquium	12 + 3	6	benotet

--	--	--	--	--

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

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

# 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  
Informatik 2  
Technische Informatik 2  
Verteilte Systeme  
Presenting and Writing in English  
Mathematik 3  
Informatik 3  
Technische Informatik 3  
Software Engineering  
Betriebssysteme  
Computernetze  
Einführung in die künstliche Intelligenz  
Datenbanksysteme

#### **Wahlpflichtbereich**

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

Quantum Information and Computation

Statistisches Lernen und Data Mining

Web-Engineering

Vertiefungspraktikum Informatik

Vertiefungsseminar Informatik

## **Wahlbereich**

Studium Generale

## **Praktische Ausbildung**

Praktische Ausbildung PO20

## **Bachelorarbeit**

Abschlussarbeit



## Titel des Moduls: English for Computer Science

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b>	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 2 Semester
<b>Lehrveranstaltungen</b> <ul style="list-style-type: none"> <li>• English for Computer Science 1 (3 CP)</li> <li>• English for Computer Science 2 (3 CP)</li> </ul>			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b>	<b>Gruppengröße</b> 25 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Einstufungstest des Zentrums für Fremdsprachenausbildung. Die Termine sind auf der folgenden Internet-Seite zu finden: <a href="http://www.zfa.ruhr-uni-bochum.de/lehre/einstufung/index.html">http://www.zfa.ruhr-uni-bochum.de/lehre/einstufung/index.html</a>		

### 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.

### 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:

- 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]

<b>Titel des Moduls: Software Engineering Praktikum</b> Software Engineering Lab					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 150 h	<b>Semester</b> 4	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Software Engineering Lab (211500)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 5 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Lernziele (learning outcomes)</b> Nach dem Modulabschluss <ul style="list-style-type: none"> <li>• können die Studierenden agil arbeiten nach SCRUM</li> <li>• wissen Studierende, wie man kleine Software-Projekte plant und können diese in Java-Android umsetzen</li> <li>• können Studierende ihre eigenen Ergebnisse in angemessener Form präsentieren</li> </ul>					
<b>Inhalt</b>  Im Software Engineering Lab wird in kleinen Projektgruppen eine Android App mit AndroidStudio entwickelt. In der begleitenden Vorlesung werden die Grundlagen moderner Softwareentwicklung vermittelt und im Projekt praktisch umgesetzt. Die Projektgruppen arbeiten selbstorganisiert agil und werden durch den gesamten Entwicklungsprozess unterstützend angeleitet. Die Entwicklung beginnt mit der Backlog-Erstellung und endet mit einem kurzen Produkt-Pitch.					
<b>Lehrformen</b> Agiles Arbeiten in Projektgruppen					
<b>Prüfungsformen</b> Projektarbeit (semesterbegleitend) mit Zwischenmeetings, Abgaben und Abschlusspräsentation					
<b>Voraussetzungen für die Vergabe von Credits</b> erfolgreich abgeschlossene Projektarbeit mit Abschlusspräsentation.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 6/165: B.Sc. Informatik [PO 20]					

<b>Titel des Moduls: Mathematik 1</b> Mathematics 1					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Mathematik 1 - Grundlagen			<b>Kontaktzeit</b> 105 h	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Erfolgreiches Bestehen der Modulklausur.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Gregor Leander Lehrende: Prof. Dr. Gregor Leander					
<b>Verwendung des Moduls</b> Bachelor Informatik  Bachelor IT-Sicherheit					
<b>Lernziele (learning outcomes)</b> 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					
<b>Inhalt</b> 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"> <li>• Grundlagen der Mathematik</li> <li>• Grundlegende mathematische Begriffe</li> <li>• Schreibweisen</li> <li>• Aussagenlogik</li> <li>• Mengenlehre</li> <li>• Relationen Algebraische Grundlagen</li> <li>• ganze Zahlen</li> <li>• Restklassen</li> <li>• Gruppen-, Ringe- und Körper-Axiome Lineare Algebra</li> <li>• Vektorräume</li> <li>• Basen</li> <li>• Dimension</li> <li>• Skalarprodukte</li> <li>• lineare Abbildungen</li> <li>• lineare Gleichungssysteme</li> <li>• Basiswechsel</li> <li>• Determinanten</li> <li>• Eigenwerttheorie</li> </ul>					
<b>Lehrformen</b> Vorlesung und Übungen					
<b>Prüfungsformen</b> 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]

<b>Titel des Moduls: Informatik 1</b> Informatics 1					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 212004 Informatik 1 (Vorlesung + Übung)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> 400 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers					
<b>Verwendung des Moduls</b> Bachelor Informatik (PO 20 + PO 22)  Bachelor IT-Sicherheit (PO 20 + PO 22)  Bachelor Angewandte Informatik (PO 20 + 22)  Bachelor Elektrotechnik und Informationstechnik (PO 20 + PO 13)					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen die Studierenden die wichtigsten Konzepte imperativer und objektorientierter Programmierung</li> <li>• können die Studierenden eigene Programme entwerfen und implementieren</li> <li>• können die Studierenden mit Grundbegriffen der Informatik wie etwa Korrektheit, Laufzeit, Boolesche Algebra, Invarianten und abstrakten Datentypen arbeiten</li> <li>• sind Studierende in der Lage, die einfachen Datenstrukturen (Arrays, Dictionaries) gezielt einzusetzen und kennen Standardalgorithmen darauf, insbesondere zum Sortieren von Arrays</li> </ul>					
<b>Inhalt</b> 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"> <li>• Imperative Programmierung (Variablen, Kontrollstrukturen, Funktionen und Rekursion, Fehlerbehandlung, Ereignisbehandlung)</li> <li>• Einfache Datenstrukturen (Array und Dictionary)</li> <li>• Objektorientierung (Klassen, Sichtbarkeit, Schnittstellen, Vererbung)</li> <li>• Einführung in eine Reihe von Informatik-Konzepten (Invarianten, Laufzeitanalyse, Sortieralgorithmen, Repräsentation von Daten im Rechner, Boolesche Algebra)</li> </ul> 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.					
<b>Lehrformen</b> Vorlesung und Übungen					
<b>Prüfungsformen</b> Klausurarbeit (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b>					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 9/165 B.Sc. Informatik					



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

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 Stunden	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Rechnerarchitektur (141142)			<b>Kontaktzeit</b> 4 SWS	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		

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

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

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

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

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

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

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

**Prüfungsformen**  
 Schriftliche Modulabschlussprüfung (120 Minuten)



**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung.

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

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

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

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

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

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

<b>Titel des Moduls: Requirements Engineering</b> Requirements Engineering					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Wintersemester (letztmalig WS 22/23)	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Requirements Engineering			<b>Kontaktzeit</b> 45 Stunden	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Martina Angela Sasse Lehrende:					
<b>Verwendung des Moduls</b> <ul style="list-style-type: none"> <li>• Bachelor Informatik (PO 20)</li> <li>• Bachelor Informatik (PO 22)</li> </ul>					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen die Studierenden die Phasen des Anforderungsprozesses und dessen Ergebnisse, die verschiedenen Akteure im Prozess und ihre jeweiligen Verantwortlichkeiten</li> <li>• werden die Studierenden in der Lage sein, verschiedene Akteure im Prozess zu identifizieren (einschließlich Entwickler und Kunden) und wissen, wie sie durch Anwendung der erlernten Methoden effektiv mit ihnen umgehen können</li> <li>• kennen die Studierenden eine Reihe von Methoden und Werkzeugen, die zur Unterstützung des Anforderungserhebungsprozesses zur Verfügung stehen, und sind in der Lage, die identifizierte Anforderungen zu verwalten und zu dokumentieren</li> <li>• werden die Studierenden sich der Herausforderung des Managements komplexer und sich ändernder Anforderungen im Kontext einer agilen Software-Entwicklung mit den erlernten Methoden und Werkzeugen stellen können</li> </ul>					
<b>Inhalt</b> Die Lehrveranstaltung „Requirements Engineering“ gibt einen systematischen Überblick über die folgenden Themengebiete: <ul style="list-style-type: none"> <li>• Requirements Engineering (Anforderungsanalyse) im Kontext unterschiedlicher Softwareentwicklungsparadigmen.</li> <li>• Wer hat welche Anforderung(en)? Identifizierung der Interessengruppen und ihrer Anforderungen (unter Verwendung einer Reihe verfügbarer Methoden, einschließlich kontextbezogener Gestaltung), Managen von Konflikten der verschiedenen Interessengruppen</li> <li>• Festlegen von Anforderungen: Personas und Anwendungsfälle, Dokumentation, Wiederverwendung von Anforderungen</li> <li>• Festlegen von funktionalen und nicht funktionalen Anforderungen (Utility vs. Usability und Sicherheit), Leistungsanforderungen und Anforderungskonflikten</li> <li>• Jetzt muss es jemand bauen: Agile Entwicklung, Dokumentation und Wiederverwendung von Anforderungen - Design Thinking: Wie dieses dem Anforderungsprozess und darüber hinaus helfen kann</li> <li>• Design Thinking: Wie dieses dem Anforderungsprozess und darüber hinaus helfen kann</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Medienunterstützung und Übungen mit Gruppenarbeit					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 90 Minuten, Präsentation des Gruppenprojekts (eine Woche vor Semesterende), individueller Reflektionsbericht max. zehn Seiten (Abgabe am Semesterende)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung sowie erfolgreiche Teilnahme am Gruppenprojekt Requirements Elicitation und Specification und bestandenen Einzel-Reflektionsbericht					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)**  
Bachelor Informatik (PO 20) : 5/165

<b>Titel des Moduls: Mathematik 2</b> Mathematics 2					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Mathematik 2 - Algorithmische Mathematik			<b>Kontaktzeit</b> 105 h	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Christian Stump Lehrende: Prof. Dr. Christian Stump					
<b>Verwendung des Moduls</b> Bachelor Informatik  Bachelor IT-Sicherheit					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende grundlegende Begriffe, Beweismethoden und Algorithmen aus der elementaren Zahlentheorie</li> <li>• können Studierende die Beweistechniken selbstständig anwenden und mathematische Sachverhalte darstellen</li> <li>• kennen Studierende erste Sätze und Methoden aus der Kombinatorik und insbesondere aus der Graphentheorie und verstehen deren strukturelle Eigenschaften</li> <li>• kennen Studierende erste fundamentale Algorithmen aus der Zahlentheorie und der Kombinatorik, können diese formalisieren, selbstständig implementieren sowie deren Laufzeiten analysieren</li> </ul>					
<b>Inhalt</b> 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.					
<b>Lehrformen</b> Vorlesung mit Übungen					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 180 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den praktischen Übungen am Rechner					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  9/165: B.Sc. Informatik [PO 22]  9/158: B.Sc. Informatik [PO 20]  9/150: B.Sc. IT-Sicherheit [PO 22]					



<b>Titel des Moduls: Informatik 2</b> <b>Computer Science 2</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> 2	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 211002: Informatik 2 - Algorithmen und Datenstrukturen			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>  keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: <a href="#">Prof. Dr. Maïke Buchin</a> Lehrende: Prof. Maïke Buchin					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit / Informationstechnik					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls: <ul style="list-style-type: none"> <li>• können Studierende Algorithmen formal beschreiben und deren Korrektheit beweisen</li> <li>• können Studierende die Laufzeit und den Speicherbedarf von Algorithmen und Datenstrukturen analysieren und bewerten</li> <li>• kennen Studierende grundlegende Datenstrukturen</li> <li>• kennen Studierende grundlegende Schemata zum Entwurf von Algorithmen sind Studierende in der Lage, Algorithmen und Datenstrukturen für spezifische Probleme zu entwickeln</li> <li>• haben die Studierenden die Grundlagen der Programmiersprache Python kennengelernt</li> </ul>					
<b>Inhalt</b> Die Vorlesung gibt einen systematischen Überblick über den Entwurf und die Analyse von Algorithmen und Datenstrukturen. Dazu werden zunächst grundlegende Methoden der Analyse (insbesondere Korrektheit, Laufzeit und Speicherbedarf) von Algorithmen vorgestellt. Anschließend werden einige Algorithmen zum Sortieren und Suchen analysiert. Ebenfalls werden verschiedene grundlegende Datenstrukturen (Listen, Felder, Suchbäume und Heaps) vorgestellt. Schließlich werden Graphen betrachtet, und zwar ihre Darstellung und diverse Algorithmen auf Graphen (Durchläufe, kürzeste Wege, minimale Spannbäume). In den Übungen lernen die Studierenden sowohl die theoretische Analyse von Algorithmen und Datenstrukturen als auch deren praktische Umsetzung in eine moderne Programmiersprache (z.B. Python).					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung und theoretische sowie praktische Übungen am Rechner.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 150 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  8/158: B.Sc. Informatik [PO 22]  8/165: B.Sc. Informatik [PO 20]  8/168: B.Sc. Angewandte Informatik [PO 22]					

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

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

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

## Titel des Moduls: Technische Informatik 2

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Digitaltechnik für ITS und Informatik (211014, ab SoSe 23)  Digitaltechnik (141304, bis SoSe 22)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Amir Moradi Lehrende: Prof. Dr. Amir Moradi					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. IT-Sicherheit					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls haben die Studierenden umfassende Kenntnisse in Boolescher Algebra, Struktur und Funktionsweise grundlegender digitaler Schaltungen, Kostenoptimierung digitaler Funktionsgruppen, Techniken zur taktsynchronen Verarbeitung von Daten, Kodierung und Verarbeitung von Daten, Struktur und Funktionsweise solcher Grundfunktionalitäten, die insbesondere in Mikroprozessorarchitekturen zentrale Bestandteile sind, erworben. Die Studierenden sind in der Lage, grundlegende Schaltungskonzepte digitaler Logik- und Funktionsblöcke zu verstehen, ihr Zusammenspiel zu analysieren, die Funktionalität zu bewerten und einfache Blöcke selbst zu entwickeln. Weiterhin werden die Bewertung und Entwicklung von mehrstufigen kombinatorischen Logikblöcken sowie von Finite State Machines (FSMs) behandelt. Die Studierenden erlernen die Hardwarebeschreibungssprache Verilog, und zu jedem Thema der Vorlesung werden Verilog-Beispiele gegeben. Die Vorlesung befasst sich ausschließlich mit (takt-)synchronen Schaltungen.					
<b>Inhalt</b> Der Kurs gibt einen systematischen Überblick über die folgenden Themen: Boolesche Algebra, Realisierung boolescher Funktionen, Minimierung boolescher Funktionen, Multiplexer, Kodierer, Dekodierer, fehlererkennende und fehlerkorrigierende Codes, Addierer, Subtrahierer, Multiplizierer, Hardwarebeschreibungssprache Verilog, Speicherelemente (Flipflops), sequentielle Schaltungen, Zähler, Schieberegister, RAM, Finite State Machines (FSMs), Timing-Analyse sequentieller Schaltungen, und kurzer Überblick über FPGAs.					
<b>Lehrformen</b> Die Vorlesung wird als seminaristischer Unterricht abgehalten, die Übungen entweder am Rechner oder mit Stift und Papier.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 120 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  5/158: B.Sc. Informatik [PO 22]  5/165: B.Sc. Informatik [PO 20]  5/150: B.Sc. IT-Sicherheit/Informationstechnik [PO 22]					





<b>Titel des Moduls: Verteilte Systeme</b> distributed systems					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Distributed Systems			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf Lehrende:					
<b>Verwendung des Moduls</b> <ul style="list-style-type: none"> <li>• Bachelor Informatik</li> <li>• Bachelor Angewandte Informatik</li> </ul>					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"> <li>• 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</li> <li>• 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</li> <li>• 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</li> <li>• 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</li> <li>• 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</li> </ul>					
<b>Inhalt</b> 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.					
<b>Lehrformen</b> Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 120 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an Übungen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> <ul style="list-style-type: none"> <li>• Bachelor Informatik (PO22) : 5/165</li> </ul>					



## Titel des Moduls: Presenting and Writing in English

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b>	<b>Turnus</b> Jedes Winter und Sommersemester	<b>Dauer</b> 2 Semester
<b>Lehrveranstaltungen</b> <ul style="list-style-type: none"><li>a) Presenting in English – Vorlesung mit integrierten Übungen (2 SWS, Wintersemester)</li><li>b) English for Computer Science 2 – Sprachkurs mit integrierten Übungen (2 SWS, Sommersemester)</li></ul>			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120	<b>Gruppengröße</b> 25 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		

### 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.

### Lernziele (learning outcomes)

Nach dem erfolgreichen Abschluss des Moduls

- können die Studierenden Fachbegriffe gezielt und effektiv in Präsentationen und schriftlichen Texten anwenden
- sind die Studierenden in der Lage, ihr Wissen über ein Fachthema bzw. eigene Erkenntnisse sicher zu vermitteln und spontan dazu Fragen zu beantworten
- sind die Studierenden in der Lage, in einer fachlichen Diskussion persönliche Standpunkte und Meinungen zu äußern und zu erfragen, Stellung zu 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:

- Sie können aus verschiedenen Quellen Informationen für ihre Präsentation zusammentragen - Sie können diese Informationen fachkonform und den Konventionen der englischen Sprache angemessen aufbereiten
- Sie sind in der Lage, die Vortragsprache (sprachliche Mittel) korrekt einzusetzen und die Präsentation publikumsadäquat vorzutragen
- Sie können auf Fragen zur Präsentation spontan eingehen
- Sie können mit fachrelevanten Vorträgen sicher umgehen.

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

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

### Inhalt

#### Zu a) Presenting in English

Der Fokus dieser Veranstaltung liegt auf der mündlichen Kompetenz, hier insbesondere im Schwerpunktbereich des Präsentierens. Im Vordergrund steht eine Erweiterung des Wortschatzes und eine Verbesserung der fachlichen Ausdrucksweise im mündlichen Kontext. Authentische Vorträge und Vorlesungen aus dem Fach

Informatik liefern sprachlichen Input und dienen als Grundlage für Analysen und als Beispiele für die Erstellung der eigenen Präsentation. Innerhalb einer Projektarbeit erhalten die Studierenden die Gelegenheit, die 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**

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

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

**Verwendung des Moduls**  
 Bachelor Informatik

**Lernziele (learning outcomes)**

Nach dem erfolgreichen Absolvieren des Moduls

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

**Inhalt**

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

Folgende Themen werden behandelt:

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

**Lehrformen**

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

**Prüfungsformen**

Schriftliche Modulabschlussprüfung über 180 Minuten

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

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

<b>Titel des Moduls: Informatik 3</b> <b>Computer Science 3</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 212002: Informatik 3 - Theoretische Informatik			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Thomas Zeume					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit / Informationstechnik					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• beherrschen die Studierenden den professionellen Umgang mit Berechnungsmodellen und ihren Beziehungen zu Sprachklassen. Dazu gehört die intellektuelle und methodische Fähigkeit, den Nachweis der Zugehörigkeit bzw. Nichtzugehörigkeit zu einer solchen Klasse zu führen.</li> <li>• ist durch Einüben von Beweistechniken wie wechselseitige Simulation oder berechenbare Reduktionen bei den Studierenden die Einsicht gereift, dass an der Oberfläche verschieden aussehende Konzepte im Kern identisch sein können. Zudem erlaubt dies den Studierenden, neue Anwendungsprobleme selbstständig zu klassifizieren.</li> <li>• haben die Studierenden mit der Turingmaschine ein einfach handhabbares Rechnermodell erlernt, das ihnen fortan als Abstraktion für alle möglichen Rechner dient.</li> <li>• haben die Studierenden fundamentale Einsichten erlangt, welche Probleme mithilfe von Rechnern effizient entschieden, zum Teil entschieden oder prinzipiell nicht entschieden werden können. Dadurch erlangen Sie ein tieferes Verständnis von der Komplexität von Berechnungsproblemen.</li> </ul>					
<b>Inhalt</b> Die Lehrveranstaltung gibt einen systematischen Überblick über die folgenden Themengebiete: <ul style="list-style-type: none"> <li>• Endliche Automaten und reguläre Ausdrücke</li> <li>• Kellerautomaten und kontextfreie Grammatiken</li> <li>• Turingmaschinen und Entscheidbarkeit</li> <li>• Nichtdeterminismus und NP-Vollständigkeitstheorie</li> </ul>					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung und Übungen, bei denen die vorgestellten Konzepte und Techniken praktisch umgesetzt werden, teilweise mit Rechnerübungen.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (150 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 8/165: B.Sc. Informatik [PO 22]  8/158: B.Sc. Informatik [PO 22]  8/168: B.Sc. Angewandte Informatik [PO 22]  8/170: B.Sc. Angewandte Informatik [PO 20]					



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

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

<b>Titel des Moduls: Technische Informatik 3</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 Stunden	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Technische Informatik 3 - Hardwareprogrammierung			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Tim Güneysu Lehrende: Prof. Dr.-Ing. Tim Güneysu					
<b>Verwendung des Moduls</b> Bachelor Informatik					
<b>Lernziele (learning outcomes)</b> Die Studierenden sollen Kenntnisse über technische Herausforderungen bei der anwendungsbezogenen Entwicklung von eingebetteten Systemen sowie des Internet of Things erlernen. Im Vordergrund der Veranstaltung stehen die maschinennahe Programmierung sowie die problemgerechte Integration von Aktorik und Sensorik. Nach dem erfolgreichen Abschluss des Moduls - kennen Studierende das Entwicklungs- und Programmiermodell sowie spezifische Eigenschaften von Mikrocontrollern als zentrale Grundlage eines eingebetteten Systems - haben Studierende die Fähigkeit zur maschinennahen Programmierung eines ausgewählten Mikrocontrollers zur Realisierung grundlegender Steuerprozesse sowie für die umgebende Peripherie 43 - erlernen Studierende Kommunikations- und Interaktionskonzepte (z.B. serielle Kommunikation via UART, SPI, PCIe etc.) mit externen Komponenten - können Studierende nebenläufige Prozesse strukturieren, in Systemen integrieren sowie damit verbundene potenzielle Probleme verstehen - sind Studierende in der Lage, komplexe anwendungsnahe Anforderungen mittels ausgewählter Hardwarekomponenten in ein eingebettetes System zu realisieren					
<b>Inhalt</b> Die Entwicklung von komplexen eingebetteten Systemen in Form einer zentralen Steuereinheit mit unterschiedlicher Sensorik und Aktorik spielen in vielen Anwendungen eine zentrale Rolle. In der Veranstaltung „Technische Informatik 3“ werden die vielfältigen Aufgabentypen und Realisierungsmöglichkeiten eines eingebetteten Systems sowie dessen anwendungsnaher Entwurf und Implementierung behandelt. Ein besonderes Gewicht in der Veranstaltung wird dabei auf ein anwendungsnahes Beispiel aus der Robotik gelegt, bei der viele der genannten Eigenschaften, Programme und Peripherie entwickelt und gesamtheitlich integriert werden müssen, um die Fähigkeiten eines einfachen Robotersystems auf vorhandener Hardware nachzubilden.					
<b>Lehrformen</b> Vorlesung (als Folien und Tafelvortrag) und Übungen, bei denen die vorgestellten Konzepte und Techniken praktisch an unterschiedlichen Hardware-Architekturen umgesetzt werden. Die Übungen beinhalten Elemente der Gruppen- und Projektarbeit					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 120 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/158: B.Sc. Informatik [PO 22]  5/165: B.Sc. Informatik [PO 20]					

<b>Titel des Moduls: Software Engineering</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 3	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Software Engineering			<b>Kontaktzeit</b>	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 350 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit/Informationstechnik					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verfügen die Studierenden über vertiefte Kenntnisse über ausgewählte Aspekte des Softwareentwicklungsprozesses</li> <li>• verfügen die Studierenden über Grundkenntnisse zum Thema Softwarequalität</li> <li>• kennen und verstehen die Studierenden die grundsätzlichen Ziele und Verantwortlichkeiten im Software-Lebenszyklus</li> <li>• kennen und verstehen die Studierenden die verschiedenen Aktivitäten innerhalb des Software-Lebenszyklus und deren Abhängigkeiten</li> <li>• sind die Studierenden in der Lage, die vermittelten Software-Entwurfsmethoden und Entwicklungsprozesse fallspezifisch anzuwenden</li> </ul>					
<b>Inhalt</b> Die Studierenden lernen unterschiedliche Formen von (klassischen und agilen) Vorgehensmodellen in der Softwareentwicklung kennen. Sie lernen Methoden der Anforderungserhebung, des Entwurfs und des Testens kennen und setzen diese in reale Fallbeispiele selbstständig um.					
<b>Lehrformen</b> Die Vorlesung wird als seminaristischer Unterricht mit Medienunterstützung abgehalten, die praktischen Übung am Rechner werden zudem weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/158: 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**

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

**Modulbeauftragte/r und hauptamtlich Lehrende**

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

**Verwendung des Moduls**

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

**Lernziele (learning outcomes)**

Nach dem erfolgreichen Absolvieren des Moduls

- erlangen die Studierenden ein solides Grundverständnis von modernen Betriebssystemen, ihrer Funktion und ihrer Implementierung
- sind die Studierenden in der Lage, verschiedene Aspekte eines Betriebssystems wie Prozess- und Speichermanagement zu verstehen und zu nutzen, sie können dabei verschiedene Designentscheidungen eigenständig analysieren und bewerten
- sind die Studierenden in der Lage, bestimmte Aspekte eines Betriebssystems selbst zu designen und diese argumentativ zu verteidigen

**Inhalt**

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

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

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

Um den Bezug zu modernen Betriebssystemen (aktuellen Versionen von Linux, Windows und macOS) herzustellen, werden die Themen an praktischen Beispielen illustriert. Dies ermöglicht es den Studierenden, die in der Vorlesung besprochenen Themen praktisch nachzuvollziehen.

**Lehrformen**

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

**Prüfungsformen**

Schriftliche Modulabschlussprüfung (90 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

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

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

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

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

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

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

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

<b>Titel des Moduls: Computernetze</b> Computer Networks					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 2	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Computernetze (211006)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 400 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Jörg Schwenk Lehrende: Dr.-Ing. Christian Mainka					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik  B.Sc. IT-Sicherheit / Informationstechnik					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende die wichtigsten Standards, die das heutige Internet verwendet.</li> <li>• kennen Studierende grundlegende Angriffskonzepte auf Computernetzwerke</li> <li>• verstehen Studierende den Zusammenhang zwischen den einzelnen Schichten eines Computernetzwerks und der darin enthaltenen Protokolle</li> <li>• können Studierende die wichtigsten Netzwerktools für Analysezwecke anwenden</li> </ul>					
<b>Inhalt</b> Die Vorlesung gibt eine Einführung in grundlegenden 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.					
<b>Lehrformen</b> Moodle-Unterstützte Hausaufgaben mit praxisnahen, computerunterstützten Übungen. Tool-der-Woche: Vorstellung, Einarbeitung, und Verwendung von Netzwerkrelevanten Computeranalysetools.					
<b>Prüfungsformen</b> schriftliche Modulabschlussprüfung von 120 min					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					





<b>Titel des Moduls: Einführung in die künstliche Intelligenz</b> Introduction to Artificial Intelligence					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b>	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Introduction to Artificial Intelligence (211045)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 250 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott, Prof. Dr. Tobias Glasmachers, Prof. Dr. Sen Cheng, Prof. Dr. Gregor Schöner, Prof. Dr. Maribel Acosta, Prof. Dr. Christian Straßer					
<b>Verwendung des Moduls</b> B.Sc. Informatik (Pflichtmodul)  B.Sc. Angewandte Informatik (Pflichtmodul)  B.Sc. IT-Sicherheit/Informationstechnik (Wahlpflichtmodul)					
<b>Lernziele (learning outcomes)</b> After successful completion of this course, students will be able to <ul style="list-style-type: none"> <li>• summarize a number of fundamental methods in artificial intelligence,</li> <li>• explain their mathematical basis and algorithmic nature,</li> <li>• apply them to simple problems,</li> <li>• decide which methods are suitable for which problems, and</li> <li>• communicate about the all that in English.</li> </ul>					
<b>Inhalt</b> This course gives an overview over representative methods in artificial intelligence: formal logic and reasoning, classical methods of AI, probabilistic reasoning, machine learning, deep neural networks, computational neuroscience, neural dynamics, perception, natural language processing, robotics.					
<b>Lehrformen</b> This course is given with the flipped/inverted classroom concept. The students work through online material beforehand and this will then be deepened in the contact sessions, which will be used for an interactive exchange between students and with the lecturer in a flexible format.					
<b>Prüfungsformen</b> Written module final exam (120 minutes)					
<b>Voraussetzungen für die Vergabe von Credits</b> passed written exam					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/158: B.Sc. Informatik [PO 22]  5/170: B.Sc. Informatik [PO 20]  5/168: B.Sc. Angewandte Informatik [PO 22]  5/170: B.Sc. Angewandte Informatik [PO 20]  5/150: B.Sc. IT-Sicherheit/Informationstechnik [PO 22]					



<b>Titel des Moduls: Datenbanksysteme</b> Database Systems					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 210 h	<b>Semester</b> 4	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Datenbanksysteme (211008)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Jun.-Prof. Dr.-Ing. Maribel Acosta Deibe Lehrende: Jun.-Prof. Dr.-Ing. Maribel Acosta Deibe					
<b>Verwendung des Moduls</b> B.Sc. Informatik [PO 22]  B.Sc. Angewandte Informatik [PO 22]					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Absolvieren des Moduls <ul style="list-style-type: none"> <li>• erlangen die Studierenden ein Grundverständnis von modernen Datenbanksystemen, ihrer Funktion und ihrer Implementierung</li> <li>• haben die Studierenden Datenmodellierungstechniken erlernt</li> <li>• haben die Studierenden die Semantik und die Syntax des Entity-Relationships Modells kennengelernt</li> <li>• kennen die Studenten das relational Datenbankmodell und die Relationale Algebra</li> <li>• kennen die Studierenden Anfragesprachen (z.B. SQL) und können diese nutzen</li> <li>• verstehen die Studierenden die Konzepte von Transaktion und Fehlerbehandlung</li> <li>• haben die Studierenden unterschiedliche Datenbankmanagementsysteme kennengelernt</li> <li>• sind die Studierenden in der Lage, neue Datenbanken zu modellieren und zu implementieren</li> <li>• haben die Studenten Kenntnisse über die Prozesse hinter einer Datenbankanfrage und wie diese optimiert werden kann</li> </ul>					
<b>Inhalt</b> Die Datenbanktechnologie ist eine Schlüsseltechnologie der praktischen und angewandten Informatik. Zentrales Thema dieser Veranstaltung sind die Modellierung, Aufbau und die Nutzung von Datenbanken. Folgende Themen werden behandelt: <ul style="list-style-type: none"> <li>• Einführung in Datenbanksysteme</li> <li>• Entity-Relationship Modell und Verbesserungen</li> <li>• das relational Datenbankmodell</li> <li>• Relationale Algebra und Kalkül</li> <li>• Die Relationale Anfragesprache SQL</li> <li>• Datenbankprogrammierung</li> <li>• physische Datenorganisation</li> <li>• Anfragebearbeitung und Optimierung</li> <li>• Transaktionsverwaltung und Fehlerbehandlung</li> </ul>					
<b>Lehrformen</b> In der wöchentlichen Vorlesung werden die Lerninhalte theoretisch vermittelt. In der 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.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					

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

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

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

<b>Titel des Moduls: Algorithmenparadigmen</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Algorithmenparadigmen (211043)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Maïke Buchin Lehrende: Prof. Dr. Maïke Buchin					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. Angewandte Informatik					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende eine Reihe von Algorithmenparadigmen</li> <li>• können Studierende basierend auf den Paradigmen effiziente Algorithmen für Probleme entwickeln</li> <li>• verstehen Studierende die Vor- und Nachteile unterschiedlicher Paradigmen</li> </ul>					
<b>Inhalt</b> In der Vorlesung betrachten wir unterschiedliche Algorithmenparadigmen, also Schemata zum Entwurf von effizienten Algorithmen. Dazu betrachten wir zunächst die bereits bekannten 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.					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung sowie Tutorien als seminaristischer Unterricht					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/170: 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]					

<b>Titel des Moduls: Autonomous Vehicles and Artificial Intelligence</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 5	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Autonomous Vehicles and Artificial Intelligence			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 25 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger, Dr. Sven Peldszus					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. IT-Sicherheit  B.Sc. Angewandte Informatik  M.Sc. Informatik  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/Informationstechnik					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"> <li>• Understanding requirements on autonomous vehicles</li> <li>• Understanding the architecture of autonomous vehicles</li> <li>• Ability to build a self-driving car with ROS2</li> <li>• Understanding and applying quality assurance for autonomous vehicles</li> </ul>					
<b>Inhalt</b> Autonomous driving is the future of individual mobility and all major manufacturers are working on fully autonomous vehicles. While there are robust and good solutions for the individual problems in autonomous driving, the main challenge lies in their integration. Altogether, an autonomous vehicle's software is the biggest problem. Therefore, the key in self-driving vehicles is about getting the software right. In this course, we will investigate the different aspects of self-driving vehicles as well as the importance and application of artificial intelligence in this domain. The course will primarily focus on the following topics: <ul style="list-style-type: none"> <li>• Requirements on autonomous vehicles</li> <li>• Architecture of autonomous vehicles</li> <li>• Operation systems and frameworks for robotic systems</li> <li>• Specification and Implementation of autonomous vehicles based on ROS2</li> <li>• Artificial intelligence for autonomous vehicles</li> <li>• Simulation of autonomous vehicles &amp;#8729; Localization and perception</li> <li>• Mission planning</li> <li>• Quality assurance for autonomous vehicles In the course's lecture, we provide the required theoretical background and practically apply the course's content in exercises by building a self-driving robot.</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Übungen					
<b>Prüfungsformen</b> Mündliche Prüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>					

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

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

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

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

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

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

5/ 97: M.Sc. Informatik

5/105: M.Sc. Angewandte Informatik

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

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

<b>Titel des Moduls: Digitale Forensik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 211017: Digitale Forensik			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan IT-Sicherheit Lehrende: Dr. Christof Fein					
<b>Verwendung des Moduls</b> B.Sc. IT-Sicherheit / Informationstechnik  B.Sc. Informatik  M.Sc. IT-Sicherheit / Netze und Systeme					
<b>Lernziele (learning outcomes)</b> 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.					
<b>Inhalt</b>  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.					
<b>Lehrformen</b> Vorlesung mit Übung als Blockveranstaltung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b>  5/150: 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]					





## **Titel des Moduls: Einführung in die Kryptographie 1**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 1	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Einführung in die Kryptographie 1 (212010)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 300 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> keine		

### **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

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

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

### **Inhalt**

Das Modul bietet einen allgemeinen Einstieg in die Funktionsweise moderner Kryptografie und Datensicherheit. Es werden grundlegende Begriffe und mathematisch/technische Verfahren der Kryptografie und der Datensicherheit erläutert. Praktisch relevante symmetrische und asymmetrische Verfahren und Algorithmen werden vorgestellt und an praxisrelevanten Beispielen erläutert.

Die Vorlesung lässt sich in zwei Teile gliedern:

Die Funktionsweise der symmetrischen Kryptographie einschließlich der Beschreibung historisch bedeutender symmetrischer Verschlüsselungsverfahren (Caesar Chiffre, Affine Chiffre) und aktueller symmetrischer Verfahren (Data Encryption Standard, Advanced Encryption Standard, Stromchiffren, One Time Pad) werden im ersten Teil behandelt.

Der zweite Teil besteht aus einer Einleitung zu asymmetrischen Verfahren und einem ihrer wichtigsten Stellvertretern (RSA). Hierzu wird eine Einführung der Grundlagen der Zahlentheorie durchgeführt, um ein grundlegendes Verständnis der Verfahren sicherzustellen (u.a. Ringe ganzer Zahlen, Gruppen, Körper, diskrete Logarithmen, euklidischer Algorithmus). Nichtsdestotrotz liegt der Schwerpunkt auf der algorithmischen Einführung des asymmetrischen Verfahrens.

### **Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Klausurarbeit (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Erfolgreiches Bestehen der Modulklausur.

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

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

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

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

<b>Titel des Moduls: Einführung in die Kryptographie 2</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 2	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Einführung in die Kryptographie 2 (211009)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 300 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Christof Paar Lehrende: Prof. Dr.-Ing. Christof Paar					
<b>Verwendung des Moduls</b> B.Sc. IT-Sicherheit  B.Sc. Informatik  B.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit / Netze und Systeme					
<b>Lernziele (learning outcomes)</b> 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.					
<b>Inhalt</b> 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.					
<b>Lehrformen</b> Vorlesung mit Übungen					
<b>Prüfungsformen</b> Klausurarbeit (120 Minuten)					

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung.

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

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

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

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

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

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

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

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

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

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

<b>Titel des Moduls: Information Theory</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Information Theory			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. IT-Sicherheit  M.Sc. Informatik  M.Sc. IT-Sicherheit / Informationstechnik  M.Sc. Angewandte Informatik					
<b>Lernziele (learning outcomes)</b> You will learn fundamental concepts, algorithms, and results in information theory. After successful completion of this course, you will know the mathematical model of information theory, how to design and analyze algorithms for a variety of information processing tasks, and how to implement them in Python. You will have independently read about a topic in information theory and presented it to your peers. You will be prepared for an advanced course or a research or thesis project in this area. Please see the course homepage for a precise list of learning objectives.					
<b>Inhalt</b> This course will give an introduction to information theory – the mathematical theory of information. Ever since its inception, information theory has had a profound impact on society. It underpins important technological developments, from reliable memories to mobile phone standards, and its versatile mathematical toolbox has found use in computer science, machine learning, physics, electrical engineering, mathematics, and many other disciplines.  Starting from probability theory, we will discuss how to mathematically model information sources and communication channels, how to optimally compress information, and how to design error-correcting codes that allow us to reliably communicate over noisy communication channels. We will also see how techniques used in information theory can be applied more generally to make predictions from noisy data.					
<b>Tentative syllabus:</b>  - Welcome, Introduction to Information Theory  - Probability Theory Refresher  - Numerical Random Variables, Convexity and Concavity, Entropy  - Symbol Codes: Lossless Compression, Huffman Algorithm  - Block Codes: Shannon's Source Coding Theorem, its Proof, and Variations  - Stream Codes: Lempel-Ziv Algorithm  - Stream Codes: Arithmetic Coding					



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

Please see the course homepage [https://qi.rub.de/it\\_ss23](https://qi.rub.de/it_ss23) for more information.

**Lehrformen**

Lecture with Exercise

**Prüfungsformen**

Wird noch nachgeliefert

**Voraussetzungen für die Vergabe von Credits**

Passed Exam.

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

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

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

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

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

5/97: M.Sc. Informatik

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

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

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Logik in der Informatik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 5	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 212013: Vorlesung + Übung			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 150 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: <a href="#">Prof. Dr. Thomas Zeume</a> Lehrende: Prof. Dr. Thomas Zeume					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc Informatik					
<b>Lernziele (learning outcomes)</b> In dieser Veranstaltung werden die formalen Grundlagen von modernen Logiken behandelt, mit einem Fokus auf ihrer Anwendung in der Informatik. Neben der klassischen Aussagenlogik und Prädikatenlogik betrachten wir auch Modallogik. Für jede dieser Logiken formalisieren wir Syntax und Semantik, lernen wie sich informatische Szenarien in ihnen modellieren lassen, und betrachten Algorithmen und Kalküle für Unerfüllbarkeit und Folgerungsbeziehung.					
<b>Inhalt</b> Logische Methoden spielen in vielen modernen Anwendungen der Informatik eine wichtige Rolle. Aus Datenbanken werden relevante Informationen mit Hilfe auf Logik basierender Anfragesprachen extrahiert; die formale Verifikation von Software und Hardware basiert auf logischen Spezifikationssprachen und Algorithmen für diese; und Methoden für das automatisierte Schlussfolgern in der künstlichen Intelligenz haben ihre Grundlage in der formalen Logik.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/170: B.Sc. Angewandte Informatik [PO 20]					

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

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

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

### **Verwendung des Moduls**

B.Sc. Informatik  
  
B.Sc. Angewandte Informatik

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

Nach dem erfolgreichen Abschluss des Moduls

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

### **Inhalt**

Moderne Hardware-Architekturen lassen sich nur durch den Einsatz nebenläufiger Programme richtig ausnutzen. Die nebenläufige Programmierung garantiert bei richtiger Anwendung eine optimale Auslastung der Hardware. Jedoch sind mit einem sorglosen Einsatz dieser Technik auch viele Risiken verbunden. Die Veranstaltung stellt Vorteile und auch Probleme nebenläufiger Programme dar und zeigt, wie sich die Performanz von Programmen verbessern lässt.

#### 1. Nebenläufigkeit: Schnelleinstieg

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

#### 2. Threads in Java

#### 3. UML-Modellierung von Nebenläufigkeit

#### 4. Neues zur Nebenläufigkeit in Java 5 und Java 6

#### 5. Realisierung von Nebenläufigkeit 6. Fortschritte Java-Konzepte für Nebenläufigkeit

#### 6. Fortschritte Java-Konzepte für Nebenläufigkeit

### **Lehrformen**

Online Vorlesung mit begleitendem eLearning Kurs

**Prüfungsformen**

Schriftliche Modulabschlussprüfung über 90 Minuten

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

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

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

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

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

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

<b>Titel des Moduls: Network Planing</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Network Planing			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Aydin Sezgin Lehrende: Prof. Aydin Sezgin					
<b>Verwendung des Moduls</b> Bachelor Informatik					
<b>Lernziele (learning outcomes)</b> Die Studierenden beherrschen die Behandlung zentraler Aspekte der Linearen Optimierung. Dies sind: <ul style="list-style-type: none"> <li>die Modellierung von Problemen im Bereich der Informationstechnik (z.B. Leistungsallokation) sowie im Alltag (z.B. Rucksackproblem, Sudoku, Ernährung) als lineare Optimierungsprobleme</li> <li>die Dualität sowie notwendige und hinreichende Bedingungen</li> <li>Verfahren, die zur effizienten Bestimmung von Lösungen führen</li> </ul>					
<b>Inhalt</b> 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. <ol style="list-style-type: none"> <li>Einleitung und Überblick <ul style="list-style-type: none"> <li>Motivation, Formulierung von linearen Problemen, Varianten, Beispiele, stückweise lineare Zielfunktionen</li> <li>Graphische Darstellung und Lösung</li> <li>Lineare Algebra: Überblick und Notation</li> </ul> </li> <li>Geometrie der linearen Optimierung <ul style="list-style-type: none"> <li>Konvexe Mengen, Polyhedra, Extrempunkte</li> </ul> </li> <li>Die Simplex-Methode <ul style="list-style-type: none"> <li>Optimalitätsbedingungen, Entwicklung, Implementierung</li> </ul> </li> <li>Dualitätstheorie <ul style="list-style-type: none"> <li>Motivation, Duales Problem, Dualitätstheorem</li> </ul> </li> <li>Spieltheorie</li> <li>Sensitivitätsanalyse (Lokale)</li> <li>Netzwerk-Fluss-Probleme <ul style="list-style-type: none"> <li>Formulierung, Probleme: Kürzester Pfad/Maximaler Fluss, Netzwerk-Simplex Algorithmus</li> </ul> </li> <li>Innere-Punkt-Methoden <ul style="list-style-type: none"> <li>Affiner Skalierungsalgorithmus</li> </ul> </li> <li>Ganzzahlige Optimierung <ul style="list-style-type: none"> <li>Formulierung</li> <li>Methoden: Branch and bound, cutting plane</li> </ul> </li> <li>Anwendungen</li> </ol>					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> 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]



<b>Titel des Moduls: Proofs are programs</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Catalin Hritcu Lehrende: Dr. Catalin Hritcu Dr. Clara Schneidewind					
<b>Verwendung des Moduls</b> o Bachelor Informatik					
<b>Lernziele (learning outcomes)</b> After successful completion of this course, students will be able to <ul style="list-style-type: none"> <li>• develop purely functional programs using recursive functions on numbers, lists, maps, and various kinds of trees, including the abstract syntax trees of programs;</li> <li>• use functional programming concepts such as type polymorphism and higher-order functions, which are increasingly becoming mainstream;</li> <li>• formally state and prove theorems in the Coq proof assistant;</li> <li>• apply different proof techniques in Coq (e.g. equational reasoning, contradiction, case analysis, induction on natural numbers, structural induction, proof automation);</li> <li>• define new inductive types and relations in Coq and prove statements about them;</li> <li>• understand the connection between constructive logics and typed functional programming that is at the heart of Coq, in which propositions are types and proofs are programs;</li> <li>• understand how the syntax and semantics of simple imperative programs can be formally defined in Coq and how to prove theorems about such programs and languages.</li> </ul>					
<b>Inhalt</b> Complex mathematical proofs on paper are difficult to write, check, and maintain. This holds not only for interesting proofs in mathematics, but also for complex formal proofs about interesting programs. For this reason, machine-checked proofs created with the help of interactive tools called proof assistants are gaining increased traction in academia and industry. Proof assistants have been used to prove the correctness and security of realistic compilers, operating systems, cryptographic libraries, or smart contracts, and also to construct machine-checked proofs for challenging mathematical results such as the four color theorem, the odd-order theorem (Feit–Thompson), or the construction of perfectoid spaces.  This course introduces the Coq proof assistant and explains how to use it to prove properties about functional programs and inductive relations. The Coq proof assistant enables us to program formal proofs interactively and it machine-checks the correctness of the proofs along the way. The design of the Coq proof assistant itself exploits a beautiful connection between programs in typed functional programming languages and proofs in constructive logics, which is known as the Curry-Howard Correspondence. This deep connection between programs and proofs should make this course interesting to both computer scientists and mathematicians. For computer scientists the goal is to demystify proofs as just programs in an elegant programming language, for which the course provides a gentle introduction. For mathematicians this course serves as an introduction to functional programming and also to the idea that proofs are not only a way to convince a human reader, but they can actually be fully formalized in a proof assistant like Coq and automatically checked by a computer.  This hands-on course is based on the Logical Foundations online textbook, which is itself formalized and machine-checked in the Coq proof assistant. The many exercises in each book chapter are to be solved weekly mostly in Coq, from easy exercises allowing the students to practice concepts from the lecture, building incrementally to slightly more interesting programs and proofs and also to various optional challenges. Finally, this course serves as the base for a more advanced course on “Foundations of Programming Languages, Verification,					

and Security”.

**Lehrformen**

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

**Prüfungsformen**

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

**Voraussetzungen für die Vergabe von Credits**

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

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

<b>Titel des Moduls: Quantum Information and Computation</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Quantum Information and Computation			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> German or Englisch (depends on audience)			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Michael Walter Lehrende: Prof. Dr. Michael Walter					
<b>Verwendung des Moduls</b> B.Sc. Informatik  B.Sc. IT-Sicherheit/Informationstechnik  M.Sc. Angewandte Informatik					
<b>Lernziele (learning outcomes)</b> 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.					
<b>Inhalt</b> This course will give an introduction to quantum information and quantum computation from the perspective of theoretical computer science. Topics to be covered will likely include: <ul style="list-style-type: none"> <li>• Fundamentals of quantum computing: quantum bits, states and operations</li> <li>• The power of quantum entanglement: nonlocal games</li> <li>• Entanglement as a resource: superdense coding and teleportation</li> <li>• Quantum circuit model of computation</li> <li>• Quantum computing with oracles: Deutsch-Jozsa, Bernstein-Vazirani, Simon</li> <li>• Quantum Fourier transform and phase estimation</li> <li>• Shor's factoring algorithm</li> <li>• Grover's search algorithm and beyond: how to solve SAT on a quantum computer?</li> <li>• From no cloning to quantum money: a peek at quantum cryptography</li> </ul> <p>The course should be of interest to students of computer science, mathematics, physics, and related disciplines. Students interested in a BSc or MSc project in quantum information, computing, cryptography, etc. are particularly encouraged to participate.</p>					
<b>Lehrformen</b> Lecture with Exercise					
<b>Prüfungsformen</b> Written Exam (120 Minutes)					
<b>Voraussetzungen für die Vergabe von Credits</b> Passed written Exam.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/158: B.Sc. Informatik [PO 22]  5/165: 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

<b>Titel des Moduls: Statistisches Lernen und Data Mining</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Statistisches Lernen und Data Mining			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Johannes Lederer Lehrende: Prof. Dr. Johannes Lederer					
<b>Verwendung des Moduls</b> B.Sc. Informatik  M.Sc. Informatik					
<b>Lernziele (learning outcomes)</b>  Nach dem erfolgreichen Abschluss des Moduls  - kennen die Studierenden Standardmethoden der Datenanalyse  - verstehen die Studierenden, wann welche Methoden passend sind  - sind die Studierenden in der Lage, die Methoden anzuwenden  - können die Studierenden die Ergebnisse interpretieren)					
<b>Inhalt</b> In dieser Lehrveranstaltung werden die grundlegenden Methoden der Datenanalyse eingeführt. Dabei werden verschiedene Datentypen berücksichtigt, insbesondere Regressionsdaten und Klassifikationsdaten. Immer werden auch die zu Grunde liegenden statistischen Modelle besprochen. Ebenfalls werden mögliche Anwendungen sowohl im Unterricht als auch in Computer-Übungen vorgestellt. 105 Ziel ist es, den gesamten Verlauf einfacher Datenanalysen zu vermitteln: Datenaufbereitung, statistische Modellbildung, Auswahl einer Methode, Implementierung der Methode, Visualisierung der Resultate und Interpretation.					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung, insbesondere Datenanalysen mit dem Computer, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend bereitgestellten Materialien und Aufgaben					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung über 90 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/158: B.Sc. Informatik [PO 22]  5/165: B.Sc. Informatik [PO 20]					

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

**Titel des Moduls: Vertiefungsseminar Informatik**

<b>Modul-Nr./Code</b>	<b>Credits</b> 3 CP	<b>Workload</b> 90 h	<b>Semester</b> 5	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> <ul style="list-style-type: none"><li>• Seminar Satisfiability (211117)</li><li>• Seminar zu Approximationsalgorithmen (211118)</li><li>• Seminar From Biological to Artificial Neural Networks (211131)</li><li>• Seminar über Grenzen in der theoretischen Informatik (211117)</li><li>• Seminar Reinforcement Learning (212119)</li><li>• Seminar Knowledge Graphs (212113)</li><li>• Machine Learning Applications (212108)</li><li>• Seminar Ressourceneffiziente Systemsoftware (212111)</li><li>• Seminar Implementation Security (212126)</li><li>• Seminar zur symmetrischen Kryptographie (212118)</li><li>• Seminar Modern Programming Languages (212115)</li><li>• Seminar Quantum Algorithms (211119)</li></ul>			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 60 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Je nach Seminarwahl: Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Informatik Lehrende: siehe jeweilige Seminare					
<b>Verwendung des Moduls</b> <ul style="list-style-type: none"><li>• Bachelor Informatik (PO 22)</li><li>• Bachelor Informatik (PO 20)</li></ul>					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"><li>• verfügen Studierende über vertiefte wissenschaftliche Kenntnisse in dem ausgewählten Seminarthema</li><li>• haben Studierende das halten eines wissenschaftlichen Vortrags praktisch eingeübt und können Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln</li><li>• können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen</li></ul>					
<b>Inhalt</b> Es werden Bachelorseminare zu mehreren relevanten Themen angeboten, wie beispielsweise zu maschinellem Lernen, Algorithmen, theoretischer Informatik oder zu Ingenieurinformatik. Von den angebotenen Themen wählen die Studierenden abhängig von den eigenen Interessen und den individuellen Vertiefungswünschen ein Thema aus. Dieses sollen die Studierenden selbstständig bearbeiten. Dazu gehören die Literaturrecherche, die Einarbeitung in das Thema und schließlich die Präsentation. Nähere Informationen sind zu den jeweiligen Seminaren im Vorlesungsverzeichnis zu entnehmen.					
<b>Lehrformen</b> Seminar					

**Prüfungsformen**

Seminarvortrag

**Voraussetzungen für die Vergabe von Credits**

Der Seminarvortrag muss mindestens mit der Note „ausreichend“ bewertet sein.

Um die Lernziele zu erreichen, besteht im Seminar Anwesenheitspflicht an mindestens 9 von 10 Terminen. Mehrfaches Fehlen muss durch ein ärztliches Attest entschuldigt werden. Die Anwesenheit beim ersten Termin ist obligatorisch, da zu diesem Termin die Themen verteilt werden. Das Seminar gilt als nicht bestanden, wenn an mehr als einem Termin unentschuldigt gefehlt wurde.

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

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

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



<b>Titel des Moduls: Web-Engineering</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 4	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Web-Engineering (128968 + 128969)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b>	<b>Gruppengröße</b> 200 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Markus König Lehrende: Prof. Dr.-Ing. Markus König Stephan Embers, M.Sc.					
<b>Verwendung des Moduls</b> B.Sc. Angewandte Informatik  B.Sc. Informatik  M.Sc. IT-Sicherheit/Informationstechnik					
<b>Lernziele (learning outcomes)</b> Die Entwicklung von Web-Anwendungen und Web-Services ist zentraler Bestandteil der Digitalisierung. Ziel der Lehrveranstaltung ist die Vermittlung von Grundlagen und bewährten Verfahren in der Web-Entwicklung. Studierende lernen konzeptuelle technologische Bausteine kennen: Transportverfahren, Webseitendarstellung, dynamische Web-Anwendungen und Web-Services. Über das konzeptuelle Verständnis hinaus werden praktische Kompetenzen vermittelt. Dazu werden moderne Werkzeuge der Web-Entwicklung, sowohl server- als auch clientseitig, vorgestellt und in den Übungssitzungen praktisch vertieft. Während der Umsetzung einfacher Web-Anwendungen stehen auch analytische Fähigkeiten im Fokus: Studierende werden befähigt, verschiedene Verfahren in Hinblick auf Performanz und Wartbarkeit zu bewerten. Diese Fähigkeiten sind in der kritischen Planungsphase von Software-Projekten unerlässlich. Nach dem erfolgreichen Abschluss des Moduls: <ul style="list-style-type: none"> <li>• kennen Studierende gängige Konzepte der Web-Entwicklung in den Aspekten Präsentation, Transport und Bereitstellung von Daten</li> <li>• beherrschen Studierende grundlegende Fähigkeiten in Webseitendarstellung, dynamischen Web-Anwendungen und modernen Services (Node.js)</li> </ul>					
<b>Inhalt</b> Im Rahmen des Modules werden den Studierenden aktuelle Techniken und Kenntnisse im Bereich der Web-Entwicklung aufgezeigt. Thematisch wird der Bereich der server- und clientseitigen Entwicklung abgedeckt. JavaScript stellt dabei eine zentrale Rolle dar. Folgende Lehrinhalte werden behandelt: <ul style="list-style-type: none"> <li>• Einführung in clientseitige Web-Entwicklung: HTML, CSS, JavaScript, Web Components</li> <li>• Transportverfahren und deren Nutzung: Representational State Transfer (REST), Asynchronous JavaScript und XML (AJAX)</li> <li>• Serverseitige Entwicklung mit Node.js und weiterführende Technologien</li> </ul>					
<b>Lehrformen</b> Hörsaalvorlesung mit Medienunterstützung, Tutorien als seminaristischer Unterricht, zusätzlich Selbststudium mit ergänzend bereitgestellten Materialien und Aufgaben.					
<b>Prüfungsformen</b> Schriftliche Modulabschlussprüfung (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene schriftliche Modulabschlussprüfung					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> 5/168: B.Sc. Angewandte Informatik [PO 22]					

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

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

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

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

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

<b>Titel des Moduls: Studium Generale</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b>	<b>Turnus</b> Jedes Semester	<b>Dauer</b> Semester
<b>Lehrveranstaltungen</b> Nichttechnische Wahlmodule Die Lehrveranstaltungen sind aus dem nichttechnischen Bereich frei wählbar.			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Abhängig von der jeweiligen Veranstaltung			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Modulbeauftragte/r: Studiendekan*in Informatik Hauptamtlich Lehrende/r: Siehe den jeweiligen Eintrag im Vorlesungsverzeichnis Lehrende:					
<b>Verwendung des Moduls</b> Unterschiedliche Studiengänge, abhängig vom gewählten Modul					
<b>Lernziele (learning outcomes)</b> 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.					
<b>Inhalt</b> 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.					
<b>Lehrformen</b> Abhängig von der gewählten Lehrveranstaltung, siehe die jeweilige Beschreibung im Vorlesungsverzeichnis					
<b>Prüfungsformen</b> Siehe den jeweiligen Eintrag im Vorlesungsverzeichnis					
<b>Voraussetzungen für die Vergabe von Credits</b>					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> Dieses Modul wird mit einem Leistungsnachweis (erfolgreich bestanden/nicht bestanden) abgeschlossen und wird nicht benotet.					

<b>Titel des Moduls: Praktische Ausbildung PO20</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 10 CP	<b>Workload</b> 300 h	<b>Semester</b> 5	<b>Turnus</b> Jedes Winter und Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> <ul style="list-style-type: none"> <li>• Industriepraktikum</li> <li>• Software-Projekt</li> <li>• Forschungsprojekt</li> </ul>			<b>Kontaktzeit</b> 15 h	<b>Selbststudium</b> 270 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch / Englisch			<b>Teilnahmevoraussetzungen</b> Abgeschlossenes Modul „Grundlagenpraktikum“		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan*in Informatik Lehrende:					
<b>Verwendung des Moduls</b> Bachelor Informatik					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls <ul style="list-style-type: none"> <li>• sind die Studierenden in der Lage, das erlernte Fachwissen anzuwenden</li> <li>• haben die Studierenden die Software-Entwicklungscompetenz maßgeblich ausgebaut, insbesondere im Hinblick auf größere Software-Systeme</li> <li>• haben die Studierenden zusätzliche Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung erworben</li> <li>• können die Studierenden eigene Lösungsstrategien erarbeiten</li> <li>• haben die Studierenden die Fähigkeit zur Arbeitsteilung und Zusammenarbeit im Team verbessert (Teamfähigkeit und Projektorganisation)</li> <li>• 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</li> <li>• haben die Studierenden die Kompetenz im Hinblick auf die Präsentation von Projektergebnissen verbessert</li> <li>• haben die Studierenden Erfahrung im Bewerbungsprozess gesammelt und sind auf das Berufsleben gut vorbereitet</li> </ul>					
<b>Inhalt</b> 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.					
<b>Lehrformen</b> Projektarbeit					
<b>Prüfungsformen</b> Projektarbeit mit tabellarischer Dokumentation					
<b>Voraussetzungen für die Vergabe von Credits</b> Erfolgreich abgeschlossenes Projekt und positiv bewertete abgegebene Dokumentation					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> Dieses Modul wird mit einem Leistungsnachweis (erfolgreich bestanden/nicht bestanden) abgeschlossen und wird nicht benotet.					

<b>Titel des Moduls: Abschlussarbeit</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 15 CP	<b>Workload</b> 450 h	<b>Semester</b>	<b>Turnus</b> Jedes Semester	<b>Dauer</b> Semester
<b>Lehrveranstaltungen</b> <ul style="list-style-type: none"> <li>a) Bachelor-Thesis (12 CP)</li> <li>b) Colloquium (3 CP)</li> </ul>			<b>Kontaktzeit</b> 15 h	<b>Selbststudium</b>	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>  Mindestens 135 absolvierte Leistungspunkte mit abgeschlossenen Module der ersten vier Semester		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Modulbeauftragte/r: Studiendekan*in Informatik Erstbetreuer*in: Jede/r am Studiengang beteiligte Hochschullehrer*in (s. Prüfungsordnung für die Regularien) Lehrende:					
<b>Verwendung des Moduls</b> <ul style="list-style-type: none"> <li>Bachelor Informatik [PO 22]</li> <li>Bachelor Informatik [PO 20]</li> </ul>					
<b>Lernziele (learning outcomes)</b> 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"> <li>Vertieftes Wissen im Bereich der bearbeiteten Aufgabenstellung</li> <li>Wissenschaftliches Arbeiten und Schreiben</li> <li>Projekt- und Zeitmanagement</li> <li>Präsentation wissenschaftlicher Ergebnisse</li> <li>Rhetorik und sprachliche Kompetenz</li> <li>Fächerübergreifendes Denken und Arbeiten</li> </ul>					
<b>Inhalt</b> 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.					
<b>Lehrformen</b> Projektarbeit					
<b>Prüfungsformen</b> Schriftliche Ausarbeitung der gestellte Aufgabe und Präsentation der Ergebnisse im Kolloquium					
<b>Voraussetzungen für die Vergabe von Credits</b> Positive Bewertung der Bachelorarbeit und des Kolloquiums sowie Teilnahme an anderen wissenschaftlichen Vorträgen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 180 ECTS)</b> <ul style="list-style-type: none"> <li>Bachelor Informatik [PO 22] :</li> <li>Bachelor Informatik [PO 20] : 15/165</li> </ul>					

