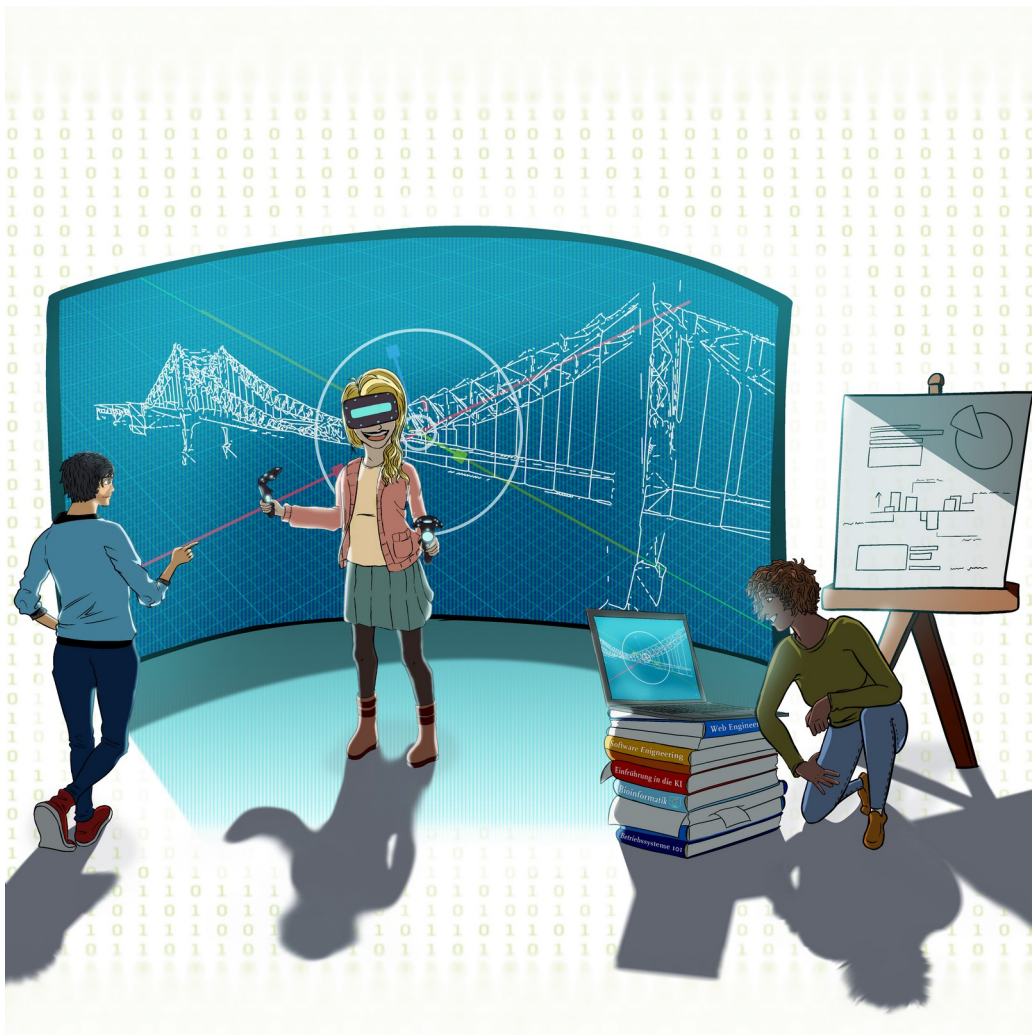


# Modulhandbuch Master of Science (M.Sc.)

## Angewandte Informatik [PO22]

Stand: Wintersemester 2025/2026



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

## Studienplan Master Angewandte Informatik Ruhr-Universität Bochum

Nr	Modul	Umfang (LP)	Empfohlenes Semester	Bewertung
<b>Wahlpflichtbereich</b>				
1	Wahlpflichtmodule**	a*	1-3	benotet
2	Vertiefungsmodule***	b*	1-3	benotet
3	Fachwissenschaftliche Vertiefung****	6	1-3	benotet
4	Studienprojekt*****	10	2-3	benotet
<b>Wahlbereich</b>				
4	Freie Wahlmodule*****	15	1-3	unbenotet
<b>Abschlussarbeit</b>				
6	Masterarbeit und Kolloquium	27+3	4	benotet
Summe:		120		

\*  $a \geq 20$ ,  $b \geq 35$ ,  $a+b \geq 59$

\*\* Hier sind Module aus dem Wahlpflichtkatalog zu belegen. Die wählbaren Module sind im jeweils aktuellen Modulhandbuch aufgeführt.

\*\*\* Hier sind Module aus den Anwendungsbereichen der Informatik zu belegen. Die wählbaren Module sind im jeweils aktuellen Modulhandbuch aufgeführt. Informationen zu den angebotenen Vertiefungsseminaren finden Sie im Vorlesungsverzeichnis der RUB.

\*\*\*\* Hier sind Vertiefungsseminare im Umfang von 6 CP zu absolvieren. Informationen zu den angebotenen Vertiefungsseminaren finden Sie im Vorlesungsverzeichnis der RUB.

\*\*\*\*\* Die wählbaren Studienprojekte werden jeweils zum Ende eines Semesters fürs Folgesemester vorgestellt.

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

**Angebotene Wahlpflicht- und Vertiefungsmodule**

Lehrveranstaltung	Einheit	Umfang Modul (LP)	Semester	Bewertung
<b>Wahlpflichtmodule</b>				
Advanced Algorithms	Informatik	9	WS	benotet
Deterministic Network Calculus	Informatik	9	WS	benotet
Fundamentals of GPU Programming	ETIT	5	WS	benotet
Kryptographie	Informatik	8	WS	benotet
Machine Learning: Unsupervised Methods	Informatik	10	WS	benotet
Numerical Optimization	Informatik	6	WS	benotet
Effiziente Algorithmen	Mathe	9	SS	benotet
Highlights of Theoretical Computer Science	Informatik	9	SS	benotet
Machine Learning: Supervised Methods	Informatik	6	SS (nicht im SS 25)	benotet
Scheduling Theory	Informatik	5	SS	benotet
Software Languages	Informatik	5	SS	benotet
Theorie des maschinellen Lernens	Informatik	9	SS	benotet
Geometrische Algorithmen (ab SS 24 Vertiefungsmodul)	Informatik	5	Letztmalig SS 23	benotet
Komplexitätstheorie	Informatik	9	Letztmalig SS 22	benotet
Information Theory (ab SS 24 Wahlpflichtmodul)	Informatik	5	Letztmalig SS 23	benotet
Machine Learning: Evolutionary Algorithms	Informatik	6	Letztmalig WS 24/25	benotet
<b>Vertiefungsmodule</b>				
<b>Ingenieurinformatik</b>				
Advanced System Engineering	MB	5	WS	benotet
Energy-Aware Computing Systems	Informatik	6	WS	benotet
Grundlagen der FEM	Bauing	5	WS	benotet
Automation in Design and Construction	Bauing	6	WS	benotet
Bildverarbeitung in der Medizin	ETIT	5	SS	benotet
Verkehrstechnik	Bauing	6	SS	benotet
Performance Engineering Praktikum	Informatik	5	Letztmalig WS 23/24	benotet
Design Optimization	Bauing	6	Letztmalig WS 22/23	benotet
<b>Programmier- und Simulationstechnik</b>				
3D-Simulation in der Automatisierungstechnik	MB	5	WS	benotet
High-Performance Computing on Clusters	Bauing	6	WS	benotet
Simulationstechnik	Bauing	5	WS	benotet
Computational Geometry (ehemals Geometrische Algorithmen)	Informatik	5	SS	benotet
High-Performance Computing on Multicore Processors (ehemals High-Performance Computing on Multi- and Manycore Processors)	Bauing	6	SS	benotet
Advanced Automatic Testing	Informatik	5	SS	benotet
Complexity Economics and Agent-based Modeling	WiWi	10	Ausgelaufen SS 25	benotet
Knowledge Graphs	Informatik	5	Letztmalig SS 23	benotet
<b>Neuroinformatik</b>				
Autonomous Vehicles and Artificial Intelligence lab	Informatik	5	WS	benotet
Deep Learning	Informatik	5	WS	benotet
Introduction to Neural Data Science	Informatik	6	WS	benotet
Autonomous Vehicles and Artificial Intelligence	Informatik	5	SS	benotet
Computational Neuroscience: Vision and Memory	Informatik	5	SS	benotet
Computational Neuroscience: Single Neuron Models	Informatik	6	SS	benotet
Master-Praktikum: Deep Learning and Natural Language Processing	Informatik	5	Unregelmäßig (nicht im WS 25/26)	benotet
Master-Praktikum: Autonomous Robotics	Informatik	3	Letztmalig SS 25	benotet
Autonomous Robotics: Action, Perception and Cognition	Informatik	6	Letztmalig SS 25	benotet
Computational Neuroscience: Neural Dynamics	Informatik	6	Letztmalig WS 24/25	benotet
<b>Kryptologie und theoretische Informatik</b>				
Formal Verification and Model Checking	Informatik	5	WS	benotet
Netzsicherheit 1	Informatik	5	WS	benotet

Quantum Information and Computation	Informatik	5	WS	benotet
Privacy Engineering, Data Governance and Usability	Informatik	5	WS	benotet
Software Security 1	Informatik	9	WS	benotet
Advanced Quantum Information and Computation	Informatik	5	SS	benotet
Netzsicherheit 2	Informatik	5	SS	benotet
Systemsicherheit	Informatik	5	SS	benotet
Mobile Network Security	Informatik	5	SS	benotet
Kryptographische Protokolle	Informatik	5	Letztmalig SS 23	benotet
Information Theory (ab SS 24 Wahlpflichtmodul)	Informatik	5	Letztmalig SS 23	benotet
Symmetrische Kryptanalyse	Informatik	5	Letztmalig WS 22/23	benotet
<b>IT in Wirtschaft , Management und Gesellschaft</b>				
KIbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz	AW + MB	5	WS	benotet
AI Ethics and Society	Informatik	5	WS	benotet
Product Lifecycle Management	MB	5	SS	benotet
Responsible AI	Informatik	6	SS	benotet
Inklusives IT-Design	AW	6	Letztmalig SS 23	benotet
<b>Computerlinguistik</b>				
Anwendungen der Computerlinguistik (Im SS 25: „Fortgeschrittene Methoden der Computerlinguistik: Theorie„ (50041) + „Fortgeschrittene Methoden der Computerlinguistik: Praxis“ (050042) oder Elicited Imitation Test: Erstellung - Durchführung – Auswertung (050045) oder Schnittstelle Informationstheorie und Informationsstatus (050046))	Philologie	10 bzw. 12	WS/SS (letztmalig SS 25)	benotet
Kernanwendungen der CL	Philologie	8	WS/SS (Nicht im WS 25/26)	benotet
NLP im Bildungssektor (im WS 25/26 050043: NLP zur Analyse von Kindertexten)	Philologie	8	WS/SS	benotet
NLP für low-resource-Daten	Philologie	8	WS/SS (Nicht im WS 25/26)	benotet
Natural Language Processing with Deep Learning	Informatik	5	WS	benotet
Engineering for Large Language Models	Informatik	6	SS	benotet
Privacy-Preserving Natural Language Processing	Informatik	6	SS	benotet
Schwerpunktseminar Computational Linguistics	Philologie	5	Letztmalig SS 24	benotet
Schwerpunktseminar Linguistic Data Science	Philologie	5	Letztmalig WS 23/24	benotet
Introduction to CL in Python	Philologie	5	Letztmalig WS 23/24	benotet
Introduction to Linguistic Data Science with R	Philologie	5	Letztmalig WS 23/24	benotet
Computational Linguistics and AI	Philologie	5	Letztmalig SS 24	benotet
Linguistic Data Science	Philologie	5	Letztmalig SS 24	benotet
<b>Bioinformatik</b>				
Master-Praktikum: Big Data in der Bioinformatik	Biologie	5	WS/SS	benotet
Artificial Intelligence in the Life Sciences	Biologie	5	WS	benotet
Bioinformatics for Proteomics 1	MPC	5	WS	benotet
Bioinformatics for Proteomics 2	MPC	5	SS	benotet
Master-Praktikum: Computational Proteomics	MPC	5	SS	benotet
Bioinformatics for Proteomics	MPC	5	Letztmalig WS 23/24	benotet
Bioimage Informatics	Biologie	5	Letztmalig WS 24/25	benotet

**Angebote Vertiefungsseminare im Modul „Fachwissenschaftliche Vertiefung“**

Lehrveranstaltung	Einheit	Umfang Modul (LP)	Semester	Bewertung
<b>Vertiefungsseminare</b>				
Seminar Computing in Engineering (ehemals Seminar Ingenieurinformatik)	BauIng	3	WS/SS	benotet
Seminar Bioinformatik	Biologie / MPC	3	WS/SS	benotet
Security Engineering	Informatik	3	WS/SS	benotet
Seminar Safety and Reliability in Artificial Intelligence	Informatik	3	WS/SS	benotet
Master-Seminar "Digitale Souveränität"	Informatik	3	WS/SS	benotet
Seminar Automated Software Engineering	Informatik	3	WS/SS	benotet
Seminar Ressourceneffiziente Systemsoftware	Informatik	3	WS/SS	benotet
Seminar Mobile Network Security	Informatik	3	WS	benotet
Recent Trends in Interpretability of Artificial Intelligence Models	Informatik	3	WS (nicht WS 25/26)	benotet
Algorithms for Decision Making	Informatik	3	WS	benotet
Seminar Algorithmen	Informatik	3	WS	benotet
Seminar Distributed Systems	Informatik	3	WS	benotet
Seminar Quantum Information and Computation (ehemals Quantum Cryptography)	Informatik	3	WS	benotet
Seminar: Introduction to Bayesian Modeling	Informatik	3	WS	benotet
Seminar Computer science perspectives on Mis- and disinformation	Informatik	3	WS	benotet
Learning meets Theoretical Computer Science	Informatik	3	WS	benotet
Master Seminar: Building Trust in Large Language Models	Informatik	3	SS	benotet
Seminar Perlen der Logik (ehemals Satisfiability)	Informatik	3	SS	benotet
Seminar Computational Neuroscience	Informatik	3	SS	benotet
Seminar From Biological to Artificial Neural Networks	Informatik	3	SS	benotet
Seminar Networked Systems	Informatik	3	SS	benotet
Seminar Mathematics and Computation	Informatik	3	SS	benotet
Seminar Quantum Algorithms	Informatik	3	SS (nicht im SS 25)	benotet
Seminar Advanced topics in Deep Learning	Informatik	3	SS	benotet
Seminar on Applied Privacy and Anonymity	Informatik	3	SS	benotet
Seminar Fortgeschrittene Themen des Model Checking	Informatik	3	Letztmalig SS 22	benotet
Seminar Implementation Security	Informatik	3	Letztmalig SS 23	benotet
Seminar on Knowledge Graphs	Informatik	3	Letztmalig WS 22/23	benotet
Seminar Topics in Deep Learning for Sequence Processing	Informatik	3	Letztmalig WS 22/23	benotet
Seminar Symmetrische Kryptanalyse	Informatik	3	Letztmalig WS 22/23	benotet
Seminar on Current Topics for Systems Security and Privacy	Informatik	3	Letztmalig WS 23/24	benotet
Security and Privacy for Mobile Systems	Informatik	3	Letztmalig WS 23/24	benotet
Perlen der theoretischen Informatik (ehemals Grenzen in der theoretischen Informatik)	Informatik	3	Letztmalig WS 23/24	benotet
Seminar Approximationsalgorithmen	Informatik	3	Letztmalig SS 24	benotet
Seminar CL (Im WS 24/25: Informationsstatus und Koreferenzauflösung (050042))	Philologie	3	Letztmalig WS 24/25	benotet

## Abkürzungen:

AW: Institut für Arbeitswissenschaften

BauIng: Fakultät für Bau- und Umweltingenieurwissenschaften

ETIT: Fakultät für Elektrotechnik und Informationstechnik

MB: Fakultät für Maschinenbau

MPC: Medizinisches Proteom Center

SS: Sommersemester

WS: Wintersemester

CP: Creditpoints

# MODULHANDBUCH

## Übersicht der Module

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

---

#### Wahlpflichtbereich

3D-Simulation in der Automatisierungstechnik  
Advanced Automatic Testing  
Advanced Quantum Information and Computation  
Advanced System Engineering  
AI Ethics And Society  
Anwendungen der Computerlinguistik [Master] (letztmalig SS 25)  
Artificial Intelligence in the Life Sciences  
Automation in Design and Construction  
Autonomous Robotics: Action, Perception, Cognition [letztmalig SS 25]  
Autonomous Vehicles and Artificial Intelligence  
Autonomous Vehicles and Artificial Intelligence Lab  
Bildverarbeitung in der Medizin  
Bioinformatics for Proteomics 1  
Bioinformatics for Proteomics 2  
Computational Geometry  
Computational Neuroscience: Single-Neuron Models  
Computational Neuroscience: Vision and Memory  
Deep Learning  
Energy-Aware Computing Systems  
Engineering for Large Language Models  
Formal Verification and Model Checking  
Grundlagen der FEM  
High-Performance Computing on Clusters  
High-Performance Computing on Multicore Processors  
Introduction to Neural Data Science  
Kernanwendungen der CL (kein Angebot im WS 25/26)  
KIbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz  
Master-Praktikum Deep Learning and Natural Language Processing (kein Angebot im SS 25)  
Master-Praktikum: Autonomous Robotics [letztmalig SS 25]  
Master-Praktikum: Big Data in der Bioinformatik  
Master-Praktikum: Computational Proteomics  
Mobile Network Security

Natural language processing with deep learning [M.Sc]  
Netzicherheit 1  
Netzicherheit 2  
NLP für low-resource-Daten (kein Angebot im WS 25/26)  
NLP im Bildungssektor  
Privacy Engineering, Data Governance and Usability  
Privacy-Preserving Natural Language Processing  
Product Lifecycle Management  
Quantum Information and Computation [M.Sc.]  
Responsible AI  
Simulationstechnik  
Software Security 1 [M.Sc.]  
Systemsicherheit  
Verkehrstechnik  
Advanced Algorithms  
Deterministic Network Calculus  
Effiziente Algorithmen  
Fundamentals of GPU Programming  
Highlights of Theoretical Computer Science [M.Sc]  
Kryptographie  
Machine Learning: Supervised Methods (kein Angebot im SS 25)  
Machine Learning: Unsupervised Methods  
Numerical Optimization  
Scheduling Theory  
Software Languages  
Theorie des maschinellen Lernens  
Fachwissenschaftliche Vertiefung (Angewandte Informatik)  
Studienprojekt (Master)

## **Wahlbereich**

Freie Wahlmodule

## **Masterarbeit**

Masterarbeit und Kolloquium (AI)

## **Titel des Moduls: 3D-Simulation in der Automatisierungstechnik**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> 3D-Simulation in der Automatisierungstechnik (139050)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Bernd Kuhlenkötter Lehrende: Dr.-Ing. Alfred Hypki					
<b>Verwendung des Modul</b> M.Sc. Angewandte Informatik M.Sc. Maschinenbau M.Sc. Sales Engineering and Product Management					
<b>Vorkenntnisse</b> Empfohlen: Besuch der Vorlesung "Grundlagen der Automatisierungstechnik"					
<b>Lernziele (learning outcomes)</b>  Nach einer allgemeinen Einführung in die Anforderungen und Möglichkeiten der 3D Simulation in der Automatisierungstechnik erlernen die Studierenden die verschiedenen Schritte, die sich von der Idee über die Modellierung der Arbeitszelle und Programmierung der Automatisierungskomponenten bis zur Virtuellen Inbetriebnahme erstrecken. Ein besonderes Augenmerk wird in dieser Vorlesung auf die Industrielle Robotik gelegt, die in zahlreichen Beispielen und Anwendungen thematisiert wird. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"><li>• beherrschen die Studierenden die Grundlagen der 3D-Simulationstechnik.</li><li>• haben die Studierenden Fähigkeit erworben, automatisierungstechnische Aufgabenstellungen vorab über eine 3D-Simulation abzubilden und abzusichern.</li><li>• kennen die Studierenden die wichtigsten Methoden und Softwaresysteme zur Lösung simulationstechnischer Probleme.</li></ul>					
<b>Inhalt</b>  Die Vorlesung deckt die folgenden Themenbereiche ab <ul style="list-style-type: none"><li>• Simulation in der Automatisierungstechnik: Anforderungen und Möglichkeiten</li><li>• Grafische 3D-Simulation</li><li>• CAD-basierte Arbeitszellenmodellierung und 3D-Datenaustausch</li><li>• Roboterprogrammierung</li><li>• Offline-Programmierung und Virtuelle Inbetriebnahme</li><li>• Grundlagen und Leistungsmerkmale von grafischen 3D-Simulationssystemen im industriellen Einsatz</li></ul> Im Bereich der Kompetenzen im Kontext Digitalisierung ist besonders die Übung zu nennen, diese besteht aus der praktischen Umsetzung der von Modellierungs-, Programmierungs- und Simulationsaufgaben mit einem kommerziell verfügbaren und industriell eingesetzten 3D- Robotersimulations- und Offline-Programmiersystem.					
<b>Lehrformen</b> Gemeinsame Vorlesung und Übung in kleineren Gruppen					

**Prüfungsformen**

Klausur '3D-Simulation in der Automatisierungstechnik' (90 Min., Anteil der Modulnote 100 %)  
Studienbegleitende Aufgaben: Umsetzung einer Simulationsaufgabe (Sofern die Aufgabe vor der Modulabschlussprüfung absolviert wird, sind optional Bonuspunkte für die Klausur möglich) (Umfang und Abgabefristen wird am Anfang des Semesters bekanntgegeben)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung: Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Advanced Algorithms</b> Advanced Algorithms					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> siehe Prüfungsordnung / see Examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Advanced Algorithms (212029)			<b>Kontaktzeit</b> 6 SWS (90 h)	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Maike Buchin Lehrende: Prof. Maike Buchin					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. Mathematik  M.Sc. IT-Sicherheit / Informationstechnik					
<b>Vorkenntnisse</b> Empfohlen: Erwartet werden grundlegende Kenntnisse zu Algorithmenentwurf und -analyse wie sie aus dem Bachelorstudium bekannt sind.					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"> <li>• Fortgeschrittene Entwurfsmethoden für Algorithmen</li> <li>• Fortgeschritten Analysemethoden für Algorithmen</li> <li>• Kenntnis weiterer Datenstrukturen und Methoden zum Entwurf von Datenstrukturen</li> <li>• Anwendung der gelernten Methoden auf neue Probleme</li> </ul>					
<b>Inhalt</b> In der Vorlesung betrachten wir fortgeschrittene Themen der Algorithmik. Nach einer kurzen Wiederholung bekannter Inhalte betrachten wir vor allem Graphalgorithmen, Approximationsalgorithmen und FPT-Algorithmen sowie exakte Algorithmen für NP-schwere Probleme. Ebenfalls betrachten wir einige neue und bekannte Datenstrukturen und deren Analyse. Die betrachteten Probleme dabei sind sowohl kombinatorisch, graphentheoretisch also auch geometrisch.					
<b>Lehrformen</b> Vorlesung (als Folien- und Tafelvortrag) und Übungen, in denen die vorgestellten Inhalte vertieft werden					
<b>Prüfungsformen</b> Mündliche (15-45 Minuten) Prüfung oder Klausur (120 Minuten) (wird zu Semesterbeginn bekannt gegeben)					
<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 120 ECTS)</b> 9/97: M.Sc. Computer Science  9/105: M.Sc. Angewandte Informatik  9/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]					



**Titel des Moduls: Advanced Automatic Testing**  
**Advanced Automatic Testing**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Advanced Automatic Testing (211067)			<b>Kontaktzeit</b> 60 h (4 SWS)	<b>Selbststudium</b> 90h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Flavio Toffalini  
 Prof. Dr. Yannic Noller  
 Lehrende: Prof. Dr. Flavio Toffalini  
 Prof. Dr. Yannic Noller

**Verwendung des Moduls**

**Vorkenntnisse**

Die Studierenden sollten über Vorkenntnisse im Bereich der System- und Softwaresicherheit verfügen. Frühere Kurse in Systemprogrammiersprachen (z. B. C und C++) und ein Mindestmaß; an gegnerischer Sicherheit.

**Lernziele (learning outcomes)**

Die Studierenden erwerben Kenntnisse über automatische Testverfahren mit Schwerpunkt auf praktischen Aspekten bei Sicherheitsproblemen. Darüber hinaus wird der Kurs die neuesten Entwicklungen auf diesem Gebiet untersuchen.

**Inhalt**

Dieser Kurs befasst sich mit den Grundlagen des automatischen Testens (Fuzzing) in der Softwaresicherheit und bietet Masterstudenten die Möglichkeit, ihre Fachkenntnisse in diesem Bereich zu vertiefen.

Der Kurs bietet einen umfassenden Überblick über automatisches Testen und behandelt grundlegende Konzepte wie White-Box-, Grey-Box- und Black-Box-Tests, Standard- und moderne Code-Explorationstechniken sowie fortgeschrittene Fehlererkennung mit logikbasierten Orakeln. Jede Lektion konzentriert sich auf einen bestimmten Aspekt der Disziplin, wodurch schrittweise ein vollständiges Verständnis des Themas aufgebaut wird und die Studenten mit den Fähigkeiten ausgestattet werden, neue Konzepte in diesem Bereich unabhängig zu erforschen.

Der Kurs deckt die folgenden Themen ab, die je nach Bedarf an die Anforderungen der Klasse angepasst werden:

- Einführung in das Fuzzing-Paradigma: Black-box-, Grey-box- und White-box-Ansätze
- Arten des Abdeckungsfeedbacks
- Heuristiken für die Codeuntersuchung (z. B. Mutatoren, Metastrategien, Seed-Auswahl, Grammatiken)
- Grundlegende Konzepte der symbolischen Ausführung und des konkollischen Testens
- Fehlererkennung und Replikation
- Fehlerorakel (z. B. Sanitizer, differenzielle Tests)
- Verwaltung von Fuzzing-Kampagnen
- Domänenspezifisches Testen (z. B. Kernel, virtuelle Geräte, Bibliotheken, IoT-Geräte)

Übungen ergänzen die Vorlesungen, indem sie praktische Erfahrungen mit den unterrichteten Prinzipien bieten. Diese Übungen sind unerlässlich für die Entwicklung des praktischen Wissens und der

Problemlösungsfähigkeiten, die für die Abschlussprüfung erforderlich sind, sowie für ein gründliches Verständnis des im Kurs vorgestellten Materials.

**Lehrformen**

Der Kurs kombiniert wöchentliche Vorlesungen und Laborsitzungen. Während der Vorlesungen werden die Studierenden ermutigt, sich mit dem Dozenten auszutauschen und sich an der Lösung einfacher Aufgaben zu beteiligen, um ihre Lernerfahrung zu verbessern.

**Prüfungsformen**

Klausur (120 Minuten). In den Übungen können Bonuspunkte erlangt werden.

**Voraussetzungen für die Vergabe von Credits**

Die Studierenden müssen die schriftliche Abschlussprüfung bestehen.  
Sie können Bonuspunkte für das Lösen der in den Übungen präsentierten Aufgaben erhalten.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit / Informationstechnik

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

<b>Titel des Moduls: Advanced Quantum Information and Computation</b> Advanced Quantum Information and Computation					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Tentatively every summer semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Advanced Quantum Information and Computation (211003)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 30 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Michael Walter, Dr. Simon Schmidt Lehrende: Prof. Dr. Michael Walter					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. IT-Sicherheit / Informationstechnik  M.Sc. IT-Sicherheit / Netze und Systeme  M.Sc. Angewandte Informatik  M.Sc. Mathematik  M.Sc. Physik					
<b>Vorkenntnisse</b> Successful participation of Quantum Information and Computation (or an equivalent course). No background in physics is required.					
<b>Lernziele (learning outcomes)</b> You will learn fundamental concepts, algorithms, and results in quantum information and computation that go beyond a first course. You will be prepared for a research or thesis project in this area.					
<b>Inhalt</b> This topical course is meant as a follow-up to our introductory course Quantum Information and Computation and is aimed at students interested in deepening their knowledge in this area. We plan to cover selected topics in quantum information and computation, e.g. how to model quantum channels, analyze nonlocal games, design quantum algorithms and cryptographic protocols, and obtain insights into which problems are easy and which are likely hard even for quantum computers. Students interested in a Bachelor's or Master's project in quantum information, computing, cryptography, etc. are particularly encouraged to participate.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Final exam (the format will depend on the number of participants).					
<b>Voraussetzungen für die Vergabe von Credits</b> Passed final exam (written exam 180 min / oral exam 15 - 45 min)					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>					

5/97: M.Sc. Computer Science

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

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

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

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

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Advanced System Engineering</b> Advanced System Engineering					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Advanced System Engineering			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Detlef Gerhard Lehrende: Prof. Dr.-Ing. Detlef Gerhard					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik M.Sc. Sales Engineering and Product Management M.Sc. Maschinenbau					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verfügen Studierende über ein breites, integriertes Wissen über die über Methoden und Werkzeuge des Advanced Systems Engineering</li> <li>• haben Studierende ein umfassendes Verständnis vom Zusammenwirken der relevanten Softwaresysteme und von Integrationskonzepten aus Sicht des Advanced Systems Engineerings erworben und können dieses situativ angepasst anwenden, um z.B. differenziert die CAx und PDM/ PLM Prozessketten zu beurteilen.</li> <li>• sind die Studierenden in der Lage Problemstellungen im Rahmen der Entwicklung komplexer technischer Systeme eigenständig zu analysieren, zu bewerten und eine Systemarchitektur auf Basis von Anforderungsmanagement zu erarbeiten.</li> <li>• verfügen Studierende über exemplarische, interdisziplinäre Methodenkompetenz der domänenübergreifenden Modellierung und können komplexe technische Systeme selbstständig in einer objektorientierten Modellierungssprache (z.B. SysML) modellieren, verstehen und die erarbeiteten Modelle bewerten</li> <li>• können kooperativ Aufgabenstellungen der Modellierung und Simulation im Umfeld von Advanced Systems Engineering in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen und über Sachverhalte in Verbindung mit dem Aufbau von IoT-Lösungen und Digitalen Zwillingen kommunizieren. Indem sie praktische Fallbeispiele und Aufgaben bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit entsprechenden Softwaresystemen auf konkrete Problemstellungen der des Modellbasierten Systems Engineering übertragen.</li> </ul>					
<b>Inhalt</b> Die Digitalisierung eröffnet zahlreiche Nutzenpotenziale für neuartige Marktleistungen der Industrie auf Basis von technischen Systemen, die durch hohe Komplexität und Interdisziplinarität gekennzeichnet sind. Ein Ansatz, die Grenzen klassischer Entwicklungsmethoden zu überwinden, ist das Advanced Systems Engineering, das auf das modellbasierte und integrative Zusammenwirken der Disziplinen und Aspekte der Produktentstehung in einem Entwicklungsansatz fokussiert und somit eine ganzheitliche Produktentstehungsmethodik unter Einbeziehung systemtechnischer Methoden darstellt. Die Vorlesung behandelt aus einer übergeordneten Perspektive des Gesamtprozesses die Methoden und Werkzeuge, die bei der Entwicklung von komplexen technischen Systemen Anwendung finden. Dabei wird insbesondere das Modell Based Systems Engineering (MBSE) Konzept sowie die					

Systemmodellierungssprache SysML angewendet und von den Studierenden selbst Projektaufgaben im Umfeld von IoT Anwendungen und dem Aufbau von Digitalen Zwillingen erarbeitet.

Folgende Themen sind Gegenstand des Moduls

- Einführung Grundlagen und Begriffe des Advanced Systems Engineering
  - Model Based Definition und Model Based Systems Engineering
  - Systemdefinition und formale Modellierung, Modellierungssprachen, z.B. Sys-ML
  - Anforderungsmanagement
  - Modellierung und Simulation zusätzlich zur mechanischen Struktur
  - Domänenübergreifende Aspekte von Steuerungstechnik/Elektrotechnik
  - Spezialisierte Modellierungsmethoden für Logik- und Verhalten
  - Lückenlose Werkzeugketten und Durchgängigkeit der Modelle
  - Integration aller Domänen in PLM (Mechanik + E/E + Software/Steuerungstechnik)
  - Simulation, Verifikation und Validierung
- 
- Methoden für die Planung, Überwachung und Durchführung von Projekten im Hinblick auf Technologie, Zeit und Kosten

#### **Lehrformen**

Übung (2 SWS) / Vorlesung (2 SWS)

#### **Prüfungsformen**

Klausur 'Advanced Systems Engineering' (90 Min., Anteil der Modulnote 100 %)

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

(Beide Prüfungsteile sind über Flexnow anzumelden)

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

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

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/150: M.Sc. Angewandte Informatik

<b>Titel des Moduls: AI Ethics And Society</b> AI Ethics And Society					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> AI Ethics And Society (212044)			<b>Kontaktzeit</b>	<b>Selbststudium</b> 105h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> The participant should have a broad interest in social sciences and humanities.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Yixin Zou, <a href="https://yixinzou.github.io/">https://yixinzou.github.io/</a> Gabriel Lima, MSc, <a href="https://thegcamilo.github.io/">https://thegcamilo.github.io/</a> Lehrende: Dr. Yixin Zou, <a href="https://yixinzou.github.io/">https://yixinzou.github.io/</a> Gabriel Lima, MSc, <a href="https://thegcamilo.github.io/">https://thegcamilo.github.io/</a>					
<b>Verwendung des Moduls</b>					
<b>Vorkenntnisse</b> Recommended but not mandatory: General understanding of machine learning and data science (e.g., from Machine Learning: Supervised Methods; or Introduction to Data Science) Responsible AI (211064) Privacy Engineering, Data Governance and Usability (212037)					
<b>Lernziele (learning outcomes)</b> After completing the course, students will be able to: <ol style="list-style-type: none"> <li>1. Critically analyze the ethical, social, and political implications of artificial intelligence (AI) through lenses other than computer science;</li> <li>2. Apply theories and concepts from the social sciences and humanities to examine sociotechnical systems;</li> <li>3. Reflect on and evaluate proposals on how to develop and deploy AI in high-risk environments.</li> </ol>					
<b>Inhalt</b> Artificial intelligence (AI) systems are now deployed in several high-risk domains, directly impacting people's lives. Numerous reports and extensive research have identified how AI systems can harm individuals and society not only after deployment, but also during their development. In this course, students will learn how AI-caused harms emerge and examine existing and proposed interventions to mitigate such harms. Rather than focusing on technical solutions to AI ethics, this course will center on perspectives from the social sciences and humanities. The course will cover the following topics: <ul style="list-style-type: none"> <li>• Discrimination, bias, and injustice: e.g., how AI disproportionately harms minoritized groups and how to develop AI equitably</li> <li>• Responsibility: e.g., who is and should be responsible when AI causes harm</li> <li>• Explainable AI (XAI): e.g., the promises and limitations of AI systems explaining their reasoning and decisions</li> <li>• The politics of AI datasets: e.g., who has a say in how datasets are developed and used to train AI systems</li> <li>• Environments costs of AI: e.g., how AI can harm the environment during its development and after deployment</li> <li>• Surveillance vs. inclusion: e.g., how AI systems are used to surveil marginalized groups under the promise of inclusion</li> <li>• Auditing: e.g., how AI systems can be audited by those being harmed by them</li> <li>• Regulation: e.g., how countries around the world propose to regulate AI</li> </ul>					

- Vision: e.g., different perspectives of how AI can be develop in the future

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Combined exam: written exam 90 min (50%) + assignments/projects (50%)

**Voraussetzungen für die Vergabe von Credits**

Passed combined exam: written exam 90 min (50%) + assignments/projects (50%)

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

**Titel des Moduls: Anwendungen der Computerlinguistik [Master] (letztmalig SS 25)**

<b>Modul-Nr./Code</b>	<b>Credits</b> 10 CP	<b>Workload</b> 300 h bzw. 360 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Wechselndes Seminarangebot (siehe unter Veranstaltungen)			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 270 h bzw. 330 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Stefanie Dipper Lehrende: Prof. Dr. Stefanie Dipper, Dr. Laarmann-Quante, u.a.					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte aus den Bachelormodulen "Einführung in die Linguistik" sowie "Methoden der Computerlinguistik". Alternativ oder ergänzend wird das Durcharbeiten von Standard-Lehrbüchern der Computerlinguistik empfohlen, z. B. von: Jurafsky, D., Martin, J. H. (2008). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Prentice-Hall.					
<b>Lernziele (learning outcomes)</b>  10 CP-Variante <ul style="list-style-type: none"><li>• Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über vertieftes Wissen und Fertigkeiten in weiteren von Ihnen ausgewählten Teilgebieten der Computerlinguistik.</li><li>• Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung. Sie kennen jeweils verschiedene Ansätze und Methoden, die in diesen Bereichen Anwendung finden, und können diese miteinander vergleichen und nach ihren Stärken und Schwächen bewerten.</li><li>• Sie können die passende Methode für eine Aufgabe auswählen und sie implementieren und evaluieren. Zudem können Sie zu Ihrer Implementation eine entsprechende wissenschaftliche Publikation verfassen.</li></ul> 12 CP-Variante: <ul style="list-style-type: none"><li>• Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über fundiertes Wissen zum aktuellen Forschungsstand in weiteren von Ihnen ausgewählten Teilgebieten der Computerlinguistik.</li><li>• Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung und kennen die jeweiligen Ansätze und Methoden. Basierend auf diesen Kenntnissen können Sie einen eigenen Ansatz entwickeln und diesen in einer Implementation umsetzen sowie evaluieren.</li><li>• Sie können zu ihrem eigenen Ansatz eine Publikation in Form eines wissenschaftlichen Aufsatzes verfassen.</li></ul>					
<b>Inhalt</b>  10 CP-Variante: In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung, Dialogmodellierung, Metaphernanalyse. In jedem Themenbereich werden Sie eigenständig mit computerlinguistischer Fachliteratur arbeiten, diese einordnen und bewerten und ausgewählte Ansätze nachimplementieren.					

12 CP-Variante:

In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem gesamten Spektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung, Dialogmodellierung, Metaphernanalyse. In jedem Teilgebiet werden Sie eigenständig mit computerlinguistischer Fachliteratur arbeiten und eigene Ansätze entwickeln und nachimplementieren.

### **Lehrformen**

Seminaristischer Unterricht mit Gruppenarbeiten, mündlichen Präsentationen von Forschungsarbeiten. Die Betreuung bei Selbststudiumanteilen von 150h+ pro Lehrveranstaltung erfolgt nach individueller Absprache. Im Regelfall erfolgen individuelle Sitzungen (Zoom oder Präsenz) im wöchentlichen oder 14-tägigen Wechsel.

### **Prüfungsformen**

Implementation und Research Thesis (semsterbegleitend)

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

Erfolgreicher Abschluss der Modulprüfung (7,0 bzw. 9 CP) sowie bestandene Studienleistung (3,0 CP) im Seminar.

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

Erlaubte Fehlzeiten in Seminaren: max. 3 Sitzungen

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

### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

10/105: M.Sc. Angewandte Informatik

12/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Artificial Intelligence in the Life Sciences**  
**Artificial Intelligence in the Life Sciences**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Artificial Intelligence in the Life Sciences (190800)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 30 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Axel Mosig Lehrende: Prof. Dr. Axel Mosig					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Programmierkenntnisse in Python, Grundlagen der Neuronalen Netze und des maschinellen Lernens					
<b>Lernziele (learning outcomes)</b> Nach Abschluss der Veranstaltung sollen die Studierenden <ul style="list-style-type: none"> <li>• die methodischen Grundlagen aus dem maschinellen Lernen verstehen, insbesondere fachspezifische Varianten von Lernalgorithmen (z.B. Anomalie-Detektion, Vision Transformer, alphafold/evoformer, Diffusionsmodelle in der Strukturvorhersage) sowie Techniken der Erklärbaren KI (Explainable AI);</li> <li>• Grundlagenwissen aus den Lebenswissenschaften erworben haben (Grundlagen der Mikroskopie, Pathologie und Strukturbiologie);</li> <li>• sich interdisziplinären Kompetenzen und Denkweisen angeeignet haben (Datenanalyse als Modellierungsproblem; Entwurf anwendungsspezifischer Validierungsmetriken);</li> <li>• Anwendungskompetenz von Deep Learning Frameworks (pytorch, STAMP und andere) in den Lebenswissenschaften erworben haben</li> <li>• Kompetenzen und Erfahrungen zur Anwendung von <i>Foundation Models</i> erworben haben; und</li> <li>• Limitierungen von KI in den Lebenswissenschaften verstehen (PAC Learnability, VC-Dimension, No-Free-Lunch Theoreme des überwachten Lernens)</li> </ul>					
<b>Inhalt</b> Auf den Gebieten der Medizin und der Biologie haben in den vergangenen Jahren Ansätze der Künstlichen Intelligenz maßgebend zum Fortschritt in der Wissenschaft und deren Anwendung beigetragen. In der Medizin sind KI-basierte Bildanalyseverfahren mittlerweile fester Bestandteil diagnostischer Verfahren, während in der Strukturbiologie die KI-basierte Vorhersage der 3D-Struktur von Proteinen mit dem alphafold-System für einen entscheidenden Durchbruch gesorgt hat. In der Vorlesung <i>Artificial Intelligence in the Life Sciences</i> werden Inhalte vermittelt, die diesen Entwicklungen zugrunde liegen und die notwendig sind, um auf diesem Gebiet zu aktuellen und neuen Entwicklungen beizutragen.  Die Vorlesung behandelt theoretische Grundlagen aus der Methodik des maschinellen Lernens ebenso wie relevante Grundlagen aus der Anwendungsdomäne, insbesondere Grundlagen der Mikroskopie und der Strukturbiologie. Als methodische Grundlage vieler KI-Systeme dienen Deep Learning Ansätze wie z.B.					

sogenannte Transformer-Architekturen oder generative neuronale Netze, unter anderem in Form der Diffusions-Modelle. Neben den eigentlich Lernalgorithmen spielen auch Techniken der Erklärbaren KI eine wichtige Rolle. Eine wichtige Schnittstelle zur Anwendungsdomäne bilden die Metriken zur Validierung. Dementsprechend werden in der Vorlesung neben den allgemein verwendeten Maßen zur Bewertung der prädiktiven Genauigkeit anwendungsspezifische Metriken diskutiert, die für eine Bewertung der Eignung und Robustheit von KI-Systemen meist unerlässlich sind. Die Grundlagen und Funktionsweisen dieser Ansätze werden in der Vorlesung präsentiert und die praktische Anwendung anhand von grundlegenden Beispielen in der Übung implementiert und erprobt. In der Vorlesung werden zudem die Limitierungen von KI-Ansätzen und deren theoretische Grundlagen (PAC-Learnability, VC-Dimension und No-Free-Lunch Theoreme) behandelt. Neben praktischen Übungsaufgaben wird das Verständnis der formalen Konzepte des maschinellen Lernens durch theoretische Übungen gefestigt.

**Lehrformen**

*Vorlesung:* in Präsenz und mit digitalem Lernmaterial (Folien, teilweise Videomaterial) mit eigenständigem Vor- und Nacharbeiten; *Praktische Übungen:* Bearbeiten von Programmieraufgaben im Selbststudium bzw. in Kleingruppen; gemeinsame Besprechung in Form von *Lab Days* während des Semesters. *Theoretische Übungen:* Lösen von Übungsaufgaben im Selbststudium bzw. in Kleingruppen; Besprechen der Aufgaben in regelmäßigen Präsenzübungen.

**Prüfungsformen**

Klausur über 120 Minuten

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Automation in Design and Construction**  
**Automation in Design and Construction**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Automation in Design and Construction (129027)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 120h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Markus König Lehrende: Prof. Dr.-Ing. Markus König, Dr.-Ing. Philipp Hagedorn					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc Bauingenieurwesen					
<b>Vorkenntnisse</b> Keine					
<b>Lernziele (learning outcomes)</b> Die Studierenden <ul style="list-style-type: none"> <li>• erwerben Kompetenzen zur Automatisierung von Prozessen im Rahmen der Planung und der Ausführung von Bauprojekten,</li> <li>• lernen Methoden zur Generierung von Bestandsmodellen,</li> <li>• setzen Softwarewerkzeuge zur simulations-basierten Planung von Logistik- und Ausführungsprozessen ein</li> <li>• erwerben Kompetenzen im Bereich der Anwendung von Sensoren, Erfassungstechnologien und Robotik auf Baustellen</li> <li>• arbeiten in Kleingruppen an ausgesuchten Problemen zur Automatisierung von Prozessen im Rahmen der Planung und der Ausführung von Bauprojekten</li> <li>• lernen aktuelle Forschungsansätze zum Einsatz von künstlicher Intelligenz im Rahmen der Planung und der Ausführung von Bauprojekten kennen</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Generierung von Bestandsmodellen</li> <li>• Einsatz von Sensoren auf der Baustelle</li> <li>• Konzepte des Lean Construction</li> <li>• Logistiksimulation</li> </ul>					

- Einsatz von Drohnen und Robotik
- Anwendung von künstlicher Intelligenz im Bauwesen

**Lehrformen**

Übung (2 SWS) / Blockseminar / Vorlesung (2 SWS)

**Prüfungsformen**

Hausarbeit 'Automation in Design and Construction' (90 Std)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung: bestandene Hausarbeit

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Autonomous Robotics: Action, Perception, Cognition [letztmalig SS 25]****Autonomous Robotics: Action, Perception, Cognition**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Autonomous Robotics: Action, Perception, Cognition (211048)			<b>Kontaktzeit</b> 3 SWS (45 h)	<b>Selbststudium</b> 135 h	<b>Gruppengröße</b> 30 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Gregor Schöner Lehrende: Prof. Dr. Gregor Schöner					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b> Der Schwerpunkt des Kurses liegt auf dem Erlernen von Konzepten und dem Üben von interdisziplinärer Wissenschaft, einschließlich Lesen und Schreiben auf wissenschaftlichem und technischem Niveau. Mathematische Konzepte werden durchgehend verwendet, daher ist das Verständnis dieser Konzepte wichtig. Mathematische Kenntnisse sind für die Beherrschung des Stoffes nicht entscheidend, aber hilfreich. Die Mathematik stammt hauptsächlich aus der qualitativen Theorie dynamischer Systeme, Attraktoren und deren Instabilitäten. Zu einigen dieser Konzepte werden kurze Tutorien angeboten.					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss: <ul style="list-style-type: none"><li>• sind die Studierenden mit den Konzepten der Theorie dynamischer Systeme vertraut und können diese praktisch anwenden,</li><li>• kennen sie die mathematischen Modelle der Bewegungserzeugung</li><li>• haben sie Übung im Lesen und Schreiben von wissenschaftlichen Arbeiten.</li></ul>					
<b>Inhalt</b> Die autonome Robotik ist ein interdisziplinäres Forschungsgebiet, in dem verkörperte Systeme, die mit eigenen Sensoren und Aktoren ausgestattet sind, ein Verhalten erzeugen, das nicht vollständig vorprogrammiert ist. Autonome Robotik umfasst somit Wahrnehmung, Bewegungsgenerierung sowie Kernelemente der Kognition wie Entscheidungsfindung, Planung und Integration mehrerer Randbedingungen. In diesem Kurs werden verschiedene Ansätze für dieses interdisziplinäre Problem behandelt. In der ersten Hälfte des Kurses liegt der Schwerpunkt auf Methoden der dynamischen Systeme zur Erzeugung von Bewegung in Fahrzeugen. Das Hauptaugenmerk des Kurses liegt jedoch auf Lösungen zur autonomen Bewegungserzeugung, die durch Analogien zur Bewegungserzeugung in Nervensystemen inspiriert sind. In der zweiten Hälfte des Kurses werden Kernprobleme der menschlichen Bewegungswissenschaft behandelt, darunter das Freiheitsgradproblem, die Koordination, die motorische Kontrolle und die Reflexsteuerung von Muskeln.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Mündliche Prüfung (30 Minuten); Übungsaufgaben für Bonuspunkte					
<b>Voraussetzungen für die Vergabe von Credits</b>					

Bestandene mündliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

**Titel des Moduls: Autonomous Vehicles and Artificial Intelligence**  
**Autonomous Vehicles and Artificial Intelligence**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Autonomous Vehicles and Artificial Intelligence (211044)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 25 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Thorsten Berger  
 Lehrende: Prof. Dr. Thorsten Berger, Dr. Sven Peldszus

**Verwendung des Moduls**

B.Sc. Informatik [bis SS 23]  
 B.Sc. IT-Sicherheit [bis SS 23]  
 B.Sc. Angewandte Informatik [bis SS 23]  
 M.Sc. Computer Science  
 M.Sc. Angewandte Informatik  
 M.Sc. IT-Sicherheit/Informationstechnik  
 M.Sc. IT-Sicherheit/Netze und Systeme [bis SS 23]

**Vorkenntnisse**

Die Vorlesung Software Engineering oder eine vergleichbare Veranstaltung, Programmiererfahrungen z.B. im Rahmen anderer Lehrveranstaltungen.

**Lernziele (learning outcomes)**

- Verständnis der Anforderungen an autonome Fahrzeuge
- Verständnis der Architektur von autonomen Fahrzeugen
- Fähigkeit, ein selbstfahrendes Auto mit ROS2 zu bauen
- Verstehen und Anwenden der Qualitätssicherung für autonome Fahrzeuge

**Inhalt**

Autonomes Fahren ist die Zukunft der individuellen Mobilität und alle großen Hersteller arbeiten an vollautonomen Fahrzeugen. Während es für die einzelnen Probleme des autonomen Fahrens robuste und gute Lösungen gibt, liegt die größte Herausforderung in deren Integration. Insgesamt stellt die Software eines autonomen Fahrzeugs das größte Problem dar. Daher liegt der Schlüssel für selbstfahrende Fahrzeuge darin, die Software richtig zu machen. In diesem Kurs werden wir die verschiedenen Aspekte von selbstfahrenden Fahrzeugen sowie die Bedeutung und Anwendung von künstlicher Intelligenz in diesem Bereich untersuchen. Der Kurs wird sich hauptsächlich auf die folgenden Themen konzentrieren:

- Anforderungen an autonome Fahrzeuge

- Architektur von autonomen Fahrzeugen
- Betriebssysteme und Frameworks für Robotersysteme
- Spezifikation und Implementierung von autonomen Fahrzeugen auf Basis von ROS2
- Künstliche Intelligenz für autonome Fahrzeuge
- Simulation von autonomen Fahrzeugen
- Lokalisierung und Wahrnehmung
- Missionsplanung
- Qualitätssicherung für autonome Fahrzeuge In der Vorlesung werden die notwendigen theoretischen Grundlagen vermittelt und die Inhalte in Übungen durch den Bau eines selbstfahrenden Roboters praktisch angewendet.

#### **Lehrformen**

Vorlesung mit Übungen

#### **Prüfungsformen**

Mündliche Modulabschlussprüfung (15-45 Minuten)

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

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

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

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

**Titel des Moduls: Autonomous Vehicles and Artificial Intelligence Lab**  
**Autonomous Vehicles and Artificial Intelligence Lab**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Autonomous Vehicles and Artificial Intelligence Lab (212035)			<b>Kontaktzeit</b> 4 SWS (60h)	<b>Selbststudium</b> 90h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> -		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr Thorsten Berger Lehrende: Prof. Dr. Thorsten Berger Dr. Sven Peldszus					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/Informationstechnik					
<b>Vorkenntnisse</b> Empfohlen: Teilnahme an der Vorlesung Autonomous Vehicles and Artificial Intelligence (211044) Programmiererfahrung in C++ oder Python (z.B. als Teil von anderen Kursen) Teilnahme an der Vorlesung Software Engineering (212000) oder einer vergleichbaren Lehrveranstaltung					
<b>Lernziele (learning outcomes)</b>  Wissen  - relevante theoretische Kenntnisse über künstliche Intelligenz und autonome Fahrzeuge erläutern können  Fertigkeiten und Fähigkeiten  - Anforderungen an autonome Fahrzeuge definieren und validieren  - eine Architektur für autonome Fahrzeuge erstellen  - ein selbstfahrendes Auto mit ROS2 bauen  - Management und Integration von künstlicher Intelligenz in komplexe, softwareintensive Systeme  - Organisation eines Teams und seines Entwicklungsprozesses für ein komplexes, softwareintensives System  - Qualitätssicherung für autonome Fahrzeuge durchführen  - Erstellen der Dokumentation des Entwicklungsprozesses und der Artefakte, die für eine Zertifizierung nach den ISO-Normen für Straßenfahrzeuge benötigt werden  - professionell mit Gruppenmitgliedern und Stakeholdern kommunizieren (in Wort und Schrift)					

## **Inhalt**

Autonomes Fahren ist die Zukunft der individuellen Mobilität, und alle großen Hersteller arbeiten an vollständig autonomen Fahrzeugen. Während es für die einzelnen Probleme des autonomen Fahrens robuste und gut erforschte Lösungen gibt, liegt die größte Herausforderung in deren Integration. Insgesamt stellt die Software eines autonomen Fahrzeugs das größte Problem dar. Daher liegt der Schlüssel für selbstfahrende Fahrzeuge darin, die Software richtig zu gestalten.

In diesem Kurs werden wir die verschiedenen Aspekte von selbstfahrenden Fahrzeugen sowie die Bedeutung und Anwendung von künstlicher Intelligenz in diesem Bereich anhand der Entwicklung eines selbstfahrenden Rennwagens praktisch studieren. Zu diesem Zweck werden die Teilnehmer mit ROS2-basierten Modellautos arbeiten. Ziel ist es, den Studierenden praktische Erfahrungen bei der Entwicklung eines autonomen Rennwagens und der Organisation des Entwicklungsprozesses zu vermitteln.

## **Lehrformen**

Die wichtigste Lernsequenz des Kurses ist ein großes Praxisprojekt. Das Projekt wird in Gruppen durchgeführt, die iterativ einen autonomen Rennwagen entwickeln und dabei theoretisches Wissen über autonomes Fahren und Softwareentwicklung anwenden und festigen. Um das Lernen zu unterstützen, basiert das Autonomous Vehicles and Artificial Intelligence Lab auf seminarähnlichen Vorlesungen, die eine Plattform für Feedback und weitere Informationen bieten. Auf der Grundlage der gesammelten Informationen aktualisieren und verfeinern die Studierenden ihre Lösungen für ein autonomes Rennauto. Die kontinuierliche Reflexion über Praxis und Theorie wird durch die laufende Erstellung eines abschließenden Projektberichts parallel zur Entwicklung des Rennwagens unterstützt, in dem die Studierenden über ihr eigenes Lernen im Kurs, die Art und Weise, wie sie und ihr Team ihren Entwicklungsprozess angehen, und ihre technischen Lösungen reflektieren. Die Studenten erhalten regelmäßiges Feedback und Anleitung, um ihr Lernen zu unterstützen.

## **Prüfungsformen**

Die Endnote wird auf der Grundlage der Teilnahme an der Entwicklung des selbstfahrenden Rennwagens, schriftlicher Projektberichte, des entwickelten autonomen Rennwagens und einer mündlichen Präsentation der Gruppenergebnisse ermittelt. Die Einzelnoten werden aus der Bewertung der individuellen und gruppenbezogenen Ergebnisse gebildet.

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

Eine aktive Beteiligung an der Entwicklung eines autonomen Rennwagens, regelmäßiger Besuch der seminarähnlichen Vorlesungen und erfolgreiche Erbringung aller Leistungen.

## **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/97: M.Sc. Computer Science

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: Bildverarbeitung in der Medizin</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Bildverarbeitung in der Medizin (141220)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b> Eine Teilnahme ist ausgeschlossen, wenn im Bachelor bereits das Modul Computersehen: Einführung (Digitale Bildverarbeitung) absolviert wurde.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Georg Schmitz Lehrende: Dr.-Ing. Stefanie Dencks					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Empfohlen: Kenntnisse der Systemtheorie, Fourier-Transformation und Signalverarbeitung, die denen entsprechen, die als Grundlagen in den Vorlesungen des Bachelorstudienganges Elektrotechnik und Informationstechnik vermittelt werden. Grundkenntnisse in der Matlab-Programmierung.					
<b>Lernziele (learning outcomes)</b>  Nach erfolgreichem Abschluss des Moduls verfügen die Studierenden über Kenntnisse der mehrdimensionalen digitalen Signalverarbeitung. Sie kennen und verstehen die Aufnahme mehrdimensionaler Bilddaten der wichtigsten diagnostischen Abbildungsverfahren, können diese modellieren und hieraus Konsequenzen für ihre Verarbeitung ableiten. Die Studierenden können die verschiedenen Schritte der Bildverarbeitung in abstrakte Aufgabenkategorien einordnen (z.B. Filterung, Segmentierung, Klassifikation) und kennen ausgewählte Verfahren im Detail und können diese erläutern und anwenden. Die Studierenden sind in der Lage, eine gegebene Bildverarbeitungsaufgabe zu analysieren, sowie einen geeigneten Lösungsweg zu entwickeln und algorithmisch umzusetzen. Die Verfahren werden am Beispiel medizinischer Bilddaten vermittelt, die Studierenden können die Verfahren aber auch auf andere Anwendungsgebiete übertragen. Durch die Übungen in Kleingruppen, teilweise an Rechnern, sind die Studierenden befähigt, das Erlernte im Team praktisch umzusetzen, Lösungsansätze zu erläutern und argumentativ zu vertreten.					
<b>Inhalt</b>  Es werden die Grundlagen und speziellen Verfahren der Bildverarbeitung vorgestellt, die insbesondere bei medizinischen Bilddaten Anwendung finden. Viele Verfahren werden jedoch auch in anderen Anwendungsfeldern wie z.B. der industriellen Bildverarbeitung eingesetzt. Im ersten Abschnitt werden sowohl die Rezeption durch das menschliche visuelle System behandelt, als auch Definitionen und Grundlagen für die Bildverarbeitung eingeführt (z.B. Diskretisierung, Abtasttheorem, globale Kenngrößen von Bildern). Der zweite Abschnitt vermittelt die wichtigsten Operationen im Ortsbereich (Histogrammmodulation, Filterung morphologische Operationen, geometrische Bildoperationen, distance transform, ...). Der dritte Abschnitt umfasst Methoden der Informationsextraktion (Segmentierung, Texturanalyse, Formbeschreibung). Im vierten Abschnitt liegt der Schwerpunkt auf der Klassifikation und verschiedenen Verfahren des Machine Learning (z.B. support vector machines, deep learning). Der fünfte Abschnitt beinhaltet die Bildrestauration. Zusätzlich wird ein Überblick über die Bildregistrierung und 3D-Visualisierung gegeben.					
<b>Lehrformen</b>					

Vorlesung mit begleitender Übung. Diese Lehrveranstaltung wird über Moodle organisiert. Die notwendigen Informationen werden in der ersten Vorlesung mitgeteilt.

**Prüfungsformen**

Mündliche Modulabschlussprüfung (15-45 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene mündliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Bioinformatics for Proteomics 1</b> Bioinformatics for Proteomics 1					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Bioinformatics for Proteomics 1 (201911)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> English			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Martin Eisenacher, Prof. Dr. Julian Uszkoreit Lehrende: Prof. Dr. Martin Eisenacher, Prof. Dr. Julian Uszkoreit					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> English and basic programming skills					
<b>Lernziele (learning outcomes)</b> <b>Discipline-specific competences:</b> After the successful completion of this module <ul style="list-style-type: none"> <li>• the students have become familiar with basic knowledge of protein biochemistry,</li> <li>• they are able to explain the principles of mass spectrometry as the key technology of proteomics,</li> <li>• they are able to explain the current methods of bioinformatics for proteomics that are used for the analysis of raw data (i.e., mass spectra) in order to identify and quantify peptides or proteins, respectively,</li> <li>• they understand the underlying algorithmic and statistical concepts of these methods,</li> <li>• they are able to apply proteomics-specific software and methods to real data</li> <li>• they are able to design and program own solution strategies in Python</li> </ul> <b>Interdisciplinary/generic competences:</b> instrumental competences: <ul style="list-style-type: none"> <li>• Intensive usage of the learning platform Moodle</li> </ul> systemic competences: <ul style="list-style-type: none"> <li>• Independent learning and working</li> <li>• Teamwork and ability to work in a team</li> </ul> communicative competences: <ul style="list-style-type: none"> <li>• Presentation of own work and results</li> <li>• Communication of bioinformatics-specific technical terms</li> <li>• Rhetoric and linguistic competence (English)</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Basics of protein biochemistry</li> <li>• Properties of amino acids</li> <li>• Basics of mass spectrometry</li> <li>• Raw data processing</li> <li>• Protein databases</li> <li>• Tryptic and in silico digest of proteins</li> </ul>					

- Principles of spectra identification search engines and scores
- Limitation of the false discovery rate using the target-decoy-approach
- PSM-specific score-correction (Percolator)
- Protein inference
- Protein quantification (label-based and label-free)
- Targeted proteomics
- Data-independent acquisition
- Quality control
- Machine Learning in Proteomics
- Software tools used in bioinformatics for proteomics (Tutorial)
- Practical (programming) tasks in Python (Tutorial)

#### **Lehrformen**

Lecture: slide-based lecture.

Tutorial: Solution of small practical exercises using real example data as homework, programming tasks, group work, live-presentation of code and software and seminar-like form of teaching.

#### **Prüfungsformen**

Oral examination or written exam

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

Passed oral examination or written exam

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Bioinformatics for Proteomics 2****Bioinformatics for Proteomics 2**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Bioinformatics for Proteomics 2 (201923)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> English			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Martin Eisenacher, Prof. Dr. Julian Uszkoreit

Lehrende: Prof. Dr. Martin Eisenacher, Prof. Dr. Julian Uszkoreit

**Verwendung des Moduls**

M.Sc. Angewandte Informatik

**Vorkenntnisse**

English and basic programming skills. Bioinformatics for Proteomics I; in the winter term (strongly recommended, but not required).

**Lernziele (learning outcomes)****Discipline-specific competences:**

After the successful completion of this module

- the students have become familiar with the most important knowledge from the lecture "Bioinformatics for Proteomics I" as a brief recapitulation,
- they have become familiar with the principles of advanced methods used in bioinformatics for proteomics,
- they are able to explain and use advanced methods that currently are employed to analyze raw data (i.e., mass spectra) and results (i.e., peptide/protein identification and quantification results) and to interpret them biologically,
- they understand the underlying algorithmic and statistical concepts of these methods,
- they are able to apply proteomics-specific software and methods to real data
- they are able to design and program own solution strategies in Python
- and they are able to apply the discussed software tools and methods to real data and problems.

**Interdisciplinary/generic competences:**

instrumental competences:

- Intensive usage of the learning platform Moodle

systemic competences:

- Independent learning and working
- Teamwork and ability to work in a team

communicative competences:

- Presentation of own work and results
- Communication of bioinformatics-specific technical terms
- Rhetoric and linguistic competence (English)

**Inhalt**

- Brief recapitulation of "Bioinformatics for Proteomics I"
- Basics of statistics
- Pre-processing of quantitative proteomics data (normalization, missing values)
- Statistics for the comparison of experimental groups (t-test, ANOVA, multiple testing correction)
- Overrepresentation and Enrichment analysis
- Metaproteomics, proteogenomics and immunopeptidomics
- Open searches, Dark matter of proteomics
- Ion mobility
- Crosslinking
- Image-MS
- Variant searches (ProtGraph)
- Improvement of the target-decoy-approach (MaCcoys)
- Software tools used in bioinformatics for proteomics (Tutorial)
- Practical (programming) tasks in Python(Tutorial)

**Lehrformen**

Lecture: slide-based lecture.

Tutorial: Solution of small practical exercises using real example data as homework, programming tasks, group work, live-presentation of code and software and seminar-like form of teaching.

**Prüfungsformen**

Oral examination or written exam

**Voraussetzungen für die Vergabe von Credits**

Passed oral examination or written exam

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Computational Geometry</b> Computational Geometry					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Computational Geometry (211056)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<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> M.Sc. Computer Science  M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Zwingend erforderlich: Grundlegende Kenntnisse über Algorithmen und Datenstrukturen. Empfohlen: Grundlagen der Stochastik.					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende grundlegende geometrische Algorithmen und Datenstrukturen</li> <li>• können Studierende Algorithmen nach dem Sweep-Paradigma analysieren und entwerfen</li> <li>• können Studierende inkrementelle Algorithmen entwerfen und analysieren, insbesondere randomisiert inkrementelle Algorithmen</li> <li>• können Studierende geometrische Algorithmen nach dem Teile-und-Herrsche Prinzip analysieren und entwerfen</li> <li>• können Studierende für Bereichsanfragen geeignete Datenstrukturen aussuchen</li> </ul>					
<b>Inhalt</b> Die Algorithmische Geometrie beschäftigt sich mit dem Entwurf und der Analyse von Algorithmen und Datenstrukturen für geometrische Probleme. Dazu werden zunächst einige grundlegende Probleme betrachtet, wie das Berechnen der konvexen Hülle einer Punktmenge, der Schnittpunkte einer Menge von Strecken oder einer Triangulierung eines einfachen Polygons. Anschließend sehen wir Algorithmen zum Berechnen bekannter geometrische Strukturen, wie das Voronoi-Diagramm, die Delaunay-Triangulierung und Arrangements. Ebenfalls betrachten wir Datenstrukturen für effiziente Anfragen auf geometrischen Daten, wie Rangetrees, kd-Bäume und Quadrees. Dabei kommen vor allem drei Arten von Algorithmen zum Einsatz: inkrementell, teile-und-herrsche, und sweep. Manche von diesen treten als randomisierte Algorithmen auf.					
<b>Lehrformen</b> Vorlesung als kombinierter Folien- und Tafelvortrag und zugehörige Übungen.					
<b>Prüfungsformen</b> Mündliche Modulabschlussprüfung (15-45 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene mündliche Modulabschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>					

5/105: M.Sc. Angewandte Informatik

5/97: M.Sc. Computer Science

**Titel des Moduls: Computational Neuroscience: Single-Neuron Models**  
**Computational Neuroscience: Single-Neuron Models**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Computational Neuroscience: Single-Neuron Models (211039)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Robert Schmidt Lehrende: Prof. Dr. Robert Schmidt					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Programmierung in Python, mathematische Kenntnisse (lineare Algebra und Infinitesimalrechnung) und Interesse an Neurobiologie.					
<b>Lernziele (learning outcomes)</b> Nach Abschluss des Moduls können Studierende <ul style="list-style-type: none"> <li>• können Studierende Techniken der Computational Neuroscience zur Simulation neuronaler Aktivität anwenden</li> <li>• haben sich Studierende mit verschiedenen Arten von Modellen einzelner Neuronen, ihrer mathematischen Beschreibung und ihren verschiedenen biologischen Abstraktionsebenen vertraut gemacht</li> <li>• haben Studierende die Fähigkeiten zur Modellierung von Neuronen, Synapsen und Schaltkreisen erworben und und können diese Modelle mit Biologie und Berechnung verknüpfen</li> <li>• besitzen Studierende ein Verständnis der biologischen Grundlage für Berechnungen in Neuronen</li> </ul>					
<b>Inhalt</b> Dieses Modul beginnt mit einer Einführung in die Neurowissenschaften und die Rolle der Computational Neuroscience. Der nächste Teil des Moduls befasst sich mit biologisch fundierten Modellen einzelner Neuronen, einschließlich Leaky-Integrate-and-Fire- und Leitwert-basierter Neuronen, aber auch mit abstrakteren Modellen neuronaler Aktivität und Spike Trains. Sie werden lernen, wie diese verschiedenen Berechnungsmodelle die zugrunde liegenden biologischen Prozesse in unterschiedlichem Maße beschreiben und vereinfachen. Wir werden im Detail untersuchen, wie diese verschiedenen Neuronenmodelle in numerischen Simulationen verwendet werden können, um Forschungsfragen zur Berechnung in einzelnen Neuronen und Schaltkreisen zu beantworten. In den vorlesungsbegleitenden Übungen werden Sie praktische Erfahrungen mit der Implementierung der verschiedenen Neuronenmodelle in Python, der Durchführung numerischer Simulationen und der Durchführung von Berechnungen im Zusammenhang mit analytischen Lösungen der Modellgleichungen und der Biophysik sammeln. Der Schwerpunkt liegt auf Einzelneuronenmodellen, aber wir werden auch verfügbare Software (z.B. NEST Desktop) nutzen, um zu untersuchen, wie Einzelneuronenmodelle in Simulationen von neuronalen Netzwerken integriert werden können. Während der Schwerpunkt des Moduls auf methodischen Fragen liegt und darauf, wie Modelle auf jeder Ebene aufgebaut, getestet und validiert werden können, werden wir auch Verbindungen zu bestimmten Gehirnregionen herstellen, um die Modelle zu motivieren und zu veranschaulichen.					
<b>Lehrformen</b> Vorlesung mit Übungen					

**Prüfungsformen**

Klausur (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestehen der Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Computational Neuroscience: Vision and Memory</b> Computational Neuroscience: Vision and Memory					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Computational Neuroscience: Vision and Memory (211049)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. Cognitive Science					
<b>Vorkenntnisse</b> Das mathematische Niveau des Kurses ist unterschiedlich, aber im Allgemeinen hoch. Das Tutorium ist fast vollständig mathematisch. Zu den erforderlichen mathematischen Grundlagen gehören Infinitesimalrechnung (Funktionen, Ableitungen, Integrale, Differentialgleichungen, ...), lineare Algebra (Vektoren, Matrizen, inneres Produkt, orthogonale Vektoren, Basissysteme, ...) und ein wenig Wahrscheinlichkeitstheorie (Wahrscheinlichkeiten, Wahrscheinlichkeitsdichten, Satz von Bayes, ...).					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss dieses Kurses <ul style="list-style-type: none"> <li>• kennen Studierende grundlegende neurobiologische Fakten über das visuelle System und den Hippocampus,</li> <li>• kennen sie eine Reihe von verwandten Modellen und Methoden der Computergestützten Neurowissenschaften,</li> <li>• verstehen sie die Mathematik dieser Methoden,</li> <li>• können über all dies in englischer Sprache kommunizieren.</li> </ul>					
<b>Inhalt</b> Diese Vorlesung behandelt die Grundlagen der Neurobiologie und Modelle der Selbstorganisation in neuronalen Systemen, insbesondere die Themen					
<b>Lernen und Selbstorganisation</b> <ul style="list-style-type: none"> <li>• Hebbsches Lernen</li> <li>• Neuronale Lerndynamik und eingeschränkte Optimierung</li> <li>• Dynamische Feldtheorie</li> </ul>					
<b>Sehen</b> <ul style="list-style-type: none"> <li>• Wahrnehmungsfelder</li> </ul>					

- Neuronale Karten
- Hippocampus
- Navigation
- Episodisches Gedächtnis
- Hopfield-Netzwerk

**Lehrformen**

Dieser Kurs wird nach dem Konzept des "flipped/inverted classroom" durchgeführt. Zunächst arbeiten die Studierenden das Online-Material selbständig durch. In der Vorlesungszeit diskutieren wir dann den Stoff, finden Verbindungen zu anderen Themen, stellen Fragen und versuchen, sie zu beantworten. Im Tutorium wird das neu erworbene Wissen in analytischen Übungen angewendet und dadurch vertieft. Ich ermutige alle Studierenden, sowohl in der Selbstlernzeit als auch im Tutorium in Teams zu arbeiten.

**Prüfungsformen**

Mündliche Modulabschlussprüfung (15-45 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene mündliche Modulabschlussprüfung.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Deep Learning</b> Deep Learning					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Deep Learning (212018)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
<b>Verwendung des Moduls</b> M.Sc. IT-Sicherheit/ Informationstechnik  M.Sc. IT-Sicherheit/ Netze und Systeme [Bis WS 22/23]  M.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b> Grundkenntnisse der Linearen Algebra und Wahrscheinlichkeitstheorie sind von Vorteil.					
<b>Lernziele (learning outcomes)</b> Die Vorlesung hat das Ziel, einen Einblick in dieses Gebiet zu vermitteln. Zu Beginn werden die grundlegenden Begriffe und Konzepte des maschinellen Lernens eingeführt. Im weiteren Verlauf wird auf verschiedene neuronale Netze, Gradienten-basierte Optimierungsverfahren und generative Modelle eingegangen.					
<b>Inhalt</b> Deep Learning ist ein Untergebiet des maschinellen Lernens, welches in den letzten Jahren zu Durchbrüchen in zahlreichen Anwendungsgebieten (wie z.B. in der Objekt- und Spracherkennung und der maschinellen Übersetzung) geführt hat. Deep Learning Methoden finden unter anderem Anwendung im Bereich IT Security.					
<b>Lehrformen</b> Vorlesung und Übung					
<b>Prüfungsformen</b> Klausur (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Klausur.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>  5/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]  5/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 20]  5/105: M.Sc. Angewandte Informatik  5/ 97: M.Sc. Computer Science					

**Titel des Moduls: Deterministic Network Calculus****Deterministic Network Calculus**

<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Deterministic Network Calculus (211054)			<b>Kontaktzeit</b> 6 SWS (90 h)	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Steffen Bondorf Lehrende: Prof. Dr. Steffen Bondorf					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. IT-Sicherheit/ Informationstechnik  M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Mathematik (Funktionsanalyse), Scheduling Theory, Computernetze					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls werden die Studierenden in der Lage sein, <ul style="list-style-type: none"><li>• komplexe, vernetzte Systeme als deterministische Warteschlangensysteme zu modellieren,</li><li>• worst-case Leistungsanalysen von bestehenden Systemen bzw. Modellen durchzuführen,</li><li>• die Herausforderungen bei der Leistungsdimensionierung von geplanten Systemen zu verstehen, dabei die Wirkungsweise zentraler Mechanismen in Computernetzen anhand des Network Calculus zu erklären,</li><li>• die vorgestellten Verfahren gegeneinander abzugrenzen und auf wissenschaftliche Fragestellungen anzuwenden.</li></ul>					
<b>Inhalt</b> Verteilte Systeme sind heutzutage allgegenwärtig, und ihre Vernetzung ist von grundlegender Bedeutung für die kontinuierliche Verbreitung und damit Verfügbarkeit von Daten. Die Bereitstellung von Daten in Echtzeit ist einer der wichtigsten nichtfunktionalen Aspekte, den sicherheitskritische Netze gewährleisten müssen. Die formale Verifizierung der Datenkommunikation im Hinblick auf die worst-case Deadlines ist grundlegend für die Zertifizierung von neu entwickelten x-by-Wire-Systemen. Diese Verifizierung erlaubt den Start von Flugzeugen, das Lenken von Autos ohne mechanische Verbindung und den Betrieb sicherheitskritischer Industrieanlagen. Daher wurden verschiedene Methoden für die worst-case Modellierung und Analyse von Echtzeitsystemen entwickelt. Eine davon ist der Deterministische Network Calculus (DNC), eine vielseitige Technik, die in verschiedenen Bereichen wie Paketvermittlung, Task Scheduling, System on Chip, softwaredefinierte Netzwerke, Netzwerke in Rechenzentren und Netzwerkvirtualisierung eingesetzt werden kann. DNC ist eine Methode zur Ableitung deterministischer Schranken für zwei der vorrangigsten Leistungsmetriken in Kommunikationssystemen: <ul style="list-style-type: none"><li>• die Ende-zu-Ende-Verzögerung von Datenflüssen und</li><li>• der Speicherplatz, den ein Server benötigt, um alle eingehenden Daten in einer Warteschlange zu puffern.</li></ul>					
<b>Lehrformen</b> Vorlesung mit Übung					

**Prüfungsformen**

Klausur (150 Minuten) oder Mündliche Modulabschlussprüfung (15-45 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur oder mündliche Modulabschlussprüfung.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

9/97: M.Sc. Computer Science

9/91: M.Sc. IT-Sicherheit/ Informationstechnik

9/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Effiziente Algorithmen</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Effiziente Algorithmen (150320 + 150321)			<b>Kontaktzeit</b> 90 h	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: PD Dr. Daniela Kacso Lehrende: PD Dr. Daniela Kacso					
<b>Verwendung des Moduls</b>  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> Die Inhalte der Veranstaltung "Datenstrukturen" bzw. "Informatik 2".					
<b>Lernziele (learning outcomes)</b>  Nach dem erfolgreichen Abschluss des Moduls:  Die Studierenden <ul style="list-style-type: none"> <li>• kennen, wählen aus und nutzen grundlegende Datenstrukturen und Graphenalgorithmen</li> <li>• sind in der Lage Analysetechniken (Korrektheitsbeweise und Laufzeitanalyse) zu erläutern und zu beurteilen</li> <li>• können auch bei praktischen Problemen entscheiden, welche der vermittelten Methoden/Algorithmen/Datenstrukturen anwendbar sind und diese nach Effizienz (insb. Laufzeit der Algorithmen) bewerten</li> <li>• können konkrete Anwendungsprobleme modellieren und bei Bedarf diese Algorithmen weiter entwickeln</li> </ul>					
<b>Inhalt</b> Die Lehrveranstaltung kann sowohl in das Gebiet der praktischen als auch in das Gebiet der theoretischen Informatik eingeordnet werden. Die zentralen Themen sind die Folgenden: <ul style="list-style-type: none"> <li>• Berechnung kürzester Pfade in Digraphen</li> <li>• Berechnung eines maximalen Flusses in einem Transportnetzwerk</li> <li>• Berechnung einer optimalen Lösung bei einem Zuordnungsproblem (auch Matching-Problem genannt)</li> </ul> Darüber hinaus beschäftigen wir uns mit Anwendungen dieser grundlegenden Probleme.					
<b>Lehrformen</b> Vortrag der Lehrenden in der Vorlesung, Gruppenarbeit in den Übungen					
<b>Prüfungsformen</b> Klausur (120 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b>  Bestandene Klausur					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

9/105: M.Sc. Angewandte Informatik

9/91: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit/ Informationstechnik [PO 22]

**Titel des Moduls: Energy-Aware Computing Systems**  
**Energy-Aware Computing Systems**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Energy-Aware Computing Systems (212030)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Timo Hönig Lehrende: Prof. Dr.-Ing. Timo Hönig					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. IT-Sicherheit/ Informationstechnik  M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Studierende, die die Vorlesung und die Übungen erfolgreich besucht haben, haben die Lernziele verfolgt und die unten aufgeführten Kompetenzen erworben. Die Studierenden können <ul style="list-style-type: none"> <li>• die Bedeutung von elektrischer Energie als Betriebsmittel für Rechensysteme verstehen</li> <li>• Trade-off-Entscheidungen im Hinblick auf ein effizientes Systemdesign (d.h. Energiebedarf vs. Leistung), insbesondere von Betriebssystemen, treffen</li> <li>• modellieren den Energiebedarf für einzelne synchrone und asynchrone Operationen</li> <li>• Strategien zur Reduzierung des Energiebedarfs für Software-Aktivitäten auf der Grundlage spezifischer Hardware-Eigenschaften (z. B. Ruhezustände) anwenden</li> <li>• Software auf kritische Abschnitte, die einen hohen Energiebedarf verursachen, analysieren.</li> </ul>					
<b>Inhalt</b> Elektrische Energie ist die wichtigste Betriebsressource für Computersysteme. Obwohl der Energiebedarf von Computern an sich eine unsichtbare Systemeigenschaft ist, sind die Auswirkungen des Energiebedarfs allgegenwärtig und in verschiedenen Erscheinungsformen offensichtlich. Als praktische Beispiele dienen plötzliche Systemausfälle (d.h. Systemzusammenbrüche) und wiederkehrende Standard-Systemoperationen (d.h. Energiemanagement). Die Vorlesung befasst sich mit dem Entwurf energiebewusster Computersysteme und konzentriert sich auf die folgenden Themen: <ul style="list-style-type: none"> <li>• Leistungs- und Energiemanagement</li> <li>• Energiebuchhaltung</li> <li>• Analyse des Energiebedarfs</li> <li>• energiebewusste Betriebssystem-Architektur</li> <li>• Hardware-Energiemanagement (z.B. DVFS, Drosselung, Ruhezustände)</li> <li>• Wärmemanagement</li> <li>• Speicher- und Dateisysteme</li> <li>• Speicherverwaltung</li> <li>• Netzwerk, drahtlose Kommunikation und Protokolle</li> <li>• Energiebewusste Server/Cluster</li> </ul>					

- Compiler-Optimierungen und Code-Umwandlung
- Anzeigetechnik
- Stromnetz

Die Vorlesung ist mit den Übungen durch Forschungsarbeiten verbunden. Die Studierenden lesen die Papiere zur Vorbereitung auf die Vorlesung. Von dort aus bilden die Forschungspapiere die Grundlage für die Diskussion und den Ausgangspunkt für die Aufgabenstellungen der Übungen. Im Rahmen der Übungen wenden die Studierenden Konzepte und Strategien aus den Forschungsarbeiten auf Systeme an und bewerten die Auswirkungen auf die Energieeffizienz des Systems.

#### **Lehrformen**

Die Vorlesung wird in Form eines Seminars abgehalten. Forschungsarbeiten zum energiebewussten Rechnen und Systemdesign werden von den Studierenden vorbereitet und in den Sitzungen diskutiert und analysiert. Zusätzlich vermittelt die Vorlesung theoretisches Wissen über grundlegende Konzepte zu den einzelnen Themen.

Im Rahmen der Übungen wenden die Studierenden ihr erworbenes Wissen an, indem sie Systemsoftware und Systemkonfigurationen zur Verbesserung der Energieeffizienz anpassen. Die Ergebnisse analysieren sie durch Leistungs- und Energiebedarfsauswertungen.

#### **Prüfungsformen**

Mündliche Modulabschlussprüfung (15-45 Minuten)

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

Bestandene Modulabschlussprüfung und erfolgreiche Teilnahme an den Übungen.

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/97: M.Sc. Informatik

6/105: M.Sc. Angewandte Informatik

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

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

<b>Titel des Moduls: Engineering for Large Language Models</b> Engineering for Large Language Models					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Engineering for Large Language Models (211065)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 120h	<b>Gruppengröße</b> 30 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Bilal Zafar Lehrende: Prof. Dr. Bilal Zafar					
<b>Verwendung des Moduls</b>					
<b>Vorkenntnisse</b> Dieser Fortgeschrittenenkurs setzt eine tiefe Vertrautheit mit grundlegenden Konzepten des Maschinellen Lernens voraus. Die Teilnehmenden sollten einen der folgenden Kurse an der RUB (oder einen ähnlichen Kurs) belegt haben: (i) Machine Learning, (ii) Deep Learning, (iii) Natural Language Processing with Deep Learning, (iv) Machine Learning: Supervised Methods. Die Teilnehmenden sollten außerdem Erfahrung mit PyTorch haben.					
<b>Lernziele (learning outcomes)</b> <ol style="list-style-type: none"> <li>1. Fähigkeit zur Feinabstimmung und Durchführung von Schlussfolgerungen mit SoTA LLMs auch bei begrenzten Ressourcen.</li> <li>2. Arbeiten mit LLMs in einer lokalen (Laptop oder PC), Compute-Cloud- und API-Umgebung.</li> </ol>					
<b>Inhalt</b> Moderne LLMs wie ChatGPT und LLaMA zeigen beeindruckende Leistungen in einer Reihe von realen Anwendungen. Ein Großteil dieser Leistung wird durch die Vergrößerung der Modellgrößen ermöglicht. Moderne Modelle bestehen aus Hunderten von Milliarden von Parametern. Diese massive Skalierung macht das Training, die Feinabstimmung und die Inferenz mit LLMs im Vergleich zu traditionellen Deep-Learning-Modellen wie CNNs und LSTMs viel schwieriger.  In diesem Kurs werden Sie die technischen Fortschritte kennenlernen, die moderne LLMs antreiben. Sie werden lernen, wie man massive LLMs betreibt, die nicht auf eine einzige GPU passen. Sie werden auch Techniken wie Caching, Quantisierung, Pruning und parameter-effizientes Finetuning kennenlernen, die eine effiziente Feinabstimmung und Inferenz ermöglichen. Sie werden auch lernen, wie Sie mit API-basierten Modellen wie GPT-4 und Claude interagieren können.					
<b>Lehrformen</b> Der Kurs wird in zwei Blöcke aufgeteilt. Der erste Block besteht aus Vorlesungen, in denen wir Ihnen den Inhalt in einer Klassenzimmerumgebung vorstellen werden. Der zweite Block besteht vollständig aus Laborsitzungen, in denen Sie an praktischen Projekten arbeiten werden.					
<b>Prüfungsformen</b> Bewertete Projektarbeit (50% der Note), E-Abschlussklausur (50% der Note) über 120 Minuten					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestehende Note in der Projektarbeit und der e-Klausur.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>					

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Fachwissenschaftliche Vertiefung (Angewandte Informatik)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> 1-3	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 2 Semester
<b>Lehrveranstaltungen</b> In jedem Semester wird eine wechselnde Auswahl an Seminaren bereitgestellt. Die zugeordneten Seminare können im Vorlesungsverzeichnis eingesehen werden.			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: siehe jeweiliges Seminar					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Abhängig von der Seminarwahl.					
<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 den 2 verschiedenen ausgewählten Seminarthemen</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> <li>• können Studierende eine schriftliche Ausarbeitung zu ihrem Seminarvortrag verfassen</li> </ul>					
<b>Inhalt</b> Es werden Vertiefungsseminare zu mehreren relevanten Themen angeboten, wie beispielsweise zu Deep Learning, zu Bioinformatik oder zu Ingenieurinformatik. Von den angebotenen Themen wählen die Studierenden abhängig von den eigenen Interessen und den individuellen Vertiefungswünschen in der Regel 2 verschiedene Seminarthemen aus. Diese sollen die Studierenden selbstständig bearbeiten. Dazu gehören die Literaturrecherche, die Einarbeitung in das Thema und schließlich die Präsentation sowie eine schriftliche Ausarbeitung. Nähere Informationen zu den jeweiligen Seminaren sind dem Vorlesungsverzeichnis zu entnehmen.  Anmerkung: Die meisten Seminare werden mit 3 CP kreditiert. Hier müssen 2 verschiedene Seminare absolviert werden (z.B. Ingenieurinformatik und Safety and Reliability Artificial Intelligence). Sollte ein Seminar mit 6 CP angeboten werden, so ist dieses ausreichend, um das Modul abzuschließen.					
<b>Lehrformen</b> Seminar					
<b>Prüfungsformen</b> Semesterbegleitend; 1-2 Seminarvorträge zu unterschiedlichen Themenbereichen und evtl. jeweils eine schriftliche Ausarbeitung.					
<b>Voraussetzungen für die Vergabe von Credits</b>					

Die meisten Seminare werden mit 3 CP kreditiert. In diesem Fall errechnet sich die Modulnote als arithmetisches Mittel aus beiden Seminarnoten. Bei Seminaren mit 6 CP ist die jeweilige Seminarnote auch die Modulnote. Die Gesamtnote muss mindestens "ausreichend" sein.

Um die Lernziele zu erreichen, besteht im Seminaren 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. Ein Seminar gilt als nicht bestanden, wenn an mehr als einem Termin unentschuldigt gefehlt wurde.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Formal Verification and Model Checking**  
**Formal Verification and Model Checking**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150h	<b>Semester</b> siehe Prüfungsordnung / see Examination Regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Formal Verification and Model Checking (212041)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 90	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Wenn bereits das frühere Modul "Model Checking" in einem vorherigen Semester absolviert wurde, ist ein Ablegen der Prüfung zum Modul "Formal Verification and Model Checking" ausgeschlossen.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Nils Jansen Lehrende: Prof. Dr. Nils Jansen					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/Informationstechnik					
<b>Vorkenntnisse</b> Grundkenntnisse der Automatentheorie.					
<b>Lernziele (learning outcomes)</b> Nach Abschluss dieses Kurses werden die Teilnehmer in der Lage sein <ul style="list-style-type: none"> <li>• Praktische Verifikationsprobleme mit modernen Werkzeugen wie SAT/SMT Solvern und Model Checkern wie PRISM oder Storm zu lösen.</li> <li>• Algorithmen zu verstehen, die modernen SAT- und SMT-Solvern wie Resolution, CDCL und CDCL(T) zugrunde liegen.</li> <li>• formale Logiken wie LTL oder CTL und die algorithmischen Implikationen der Modellprüfung anhand solcher Eigenschaften zu verstehen.</li> <li>• SAT- und SMT-Methoden mit Problemen aus der klassischen Planung, der Programmverifikation, der Modellprüfung und der probabilistischen Inferenz zu verbinden.</li> <li>• Situationen zu erkennen, in denen die Anwendung von Model Checking und Verifikationstechniken für Spezifikation und Analyse nützlich sein kann.</li> </ul> <p>Arbeiten Sie an innovativer Forschung, die maschinelles Lernen und künstliche Intelligenz mit den strengen Techniken der formalen Verifikation kombiniert.</p>					
<b>Inhalt</b>  Komplexe digitale Systeme sind in unserem Leben immer präsenter und wirken sich immer stärker aus, so dass die formale Überprüfung der Korrektheit dieser Systeme von entscheidender Bedeutung ist. So darf beispielsweise das Programm eines Raumschiffs nicht abstürzen, und ein Netzwerksystem sollte auch dann noch funktionieren, wenn ein Server ausfällt. Um dies zu gewährleisten, ist ein Ansatz die deduktive Überprüfung, bei der eine Sammlung von logischen Bedingungen erstellt wird, die das System erfüllen muss. Diese Bedingungen können mit automatischen Theorembeweisern wie SAT- und SMT-Solvern algorithmisch verifiziert werden. Ein anderer beliebter Ansatz ist die Modellprüfung, die darin besteht, das System durch mathematische Modelle darzustellen und bestimmte Eigenschaften, die oft in temporalen Logiken beschrieben werden, erschöpfend zu					

prüfen. In diesem Kurs werden verschiedene Aspekte der formalen Verifikation und der Modellprüfung behandelt, darunter:

- Erfüllbarkeit von Sätzen: Auflösung und die wichtigsten Bestandteile moderner Erfüllbarkeitsbeweiser.
- Satisfiability modulo theories, insbesondere unter Verwendung linearer Ungleichungen, und der zugrunde liegende Simplex-Algorithmus.  
Explizite Zustands- und symbolische Algorithmen für die Modellprüfung von temporalen Logiken mit linearer Zeit (LTL) und Verzweigungszeit (CTL) für endliche Maschinen.
- Symbolische Modellprüfung unter Verwendung von BDDs.
- Modellprüfung unter Wahrscheinlichkeiten und Ungewissheit, unter Verwendung von Markov-Ketten und Markov-Entscheidungsprozessen, mit der PCTL-Logik und den Eigenschaften der erwarteten Belohnung
- Die Technik des maschinellen Lernens (Reinforcement Learning) sowie Techniken der Modellprüfung und der formalen Verifikation, um ihre Anwendung auf kritischen Systemen sicher zu machen

#### **Lehrformen**

- 90 Minuten wöchentliche Vorlesung, vor Ort.
- 90-minütige Übungsstunde, vor Ort.
- Wöchentliche Übungen, sowohl praktisch als auch theoretisch

#### **Prüfungsformen**

50% Klausur (90 Minuten) + 50% Projektarbeit

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

Projekt und Modulabschlussprüfung bestanden.

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. IT-Sicherheit/Informationstechnik

<b>Titel des Moduls: Fundamentals of GPU Programming</b> Fundamentals of GPU Programming					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Fundamentals of GPU Programming (141374)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Denis Eremin Lehrende: Dr. Denis Eremin					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> C (Programmiersprache)					
<b>Lernziele (learning outcomes)</b> Die Studierenden erlernen das Programmieren auf Grafikprozessoren (GPUs)					
<b>Inhalt</b> Zu einem bestimmten Zeitpunkt um 2003 stieg die Rechenleistung nicht auf Kosten der Taktfrequenz des Prozessors, sondern durch Erhöhung der Anzahl der auf dem Prozessorchip zugewiesenen Rechenkern. Grafikprozessoren (GPUs) sind die Meister dieser Computer-Hardware-Entwicklung und bieten bis zu Zehntausende einzelner Kerneinheiten. Gleichzeitig wird das GPU-Speichersystem nicht so sehr durch die Kompatibilitätsanforderungen mit älteren Generationen eingeschränkt wie CPU-Speichersysteme. Deswegen zeigen GPUs im Vergleich zu ihren älteren "Bruder" -Zentraleinheiten (CPUs) eine deutlich bessere Rohleistung der Recheneinheiten und des Speichersystems. Ursprünglich für Videobearbeitungsaufgaben entwickelt, wird die enorme Rechenleistung moderner GPUs üblicherweise zur Unterstützung von CPUs oder zur Lösung einer Vielzahl von Rechenproblemen mit (massiv) parallelisierbaren Teilen verwendet, wodurch Teraflops-hohe Rechenleistung kann schon auf Laptop- / Desktop-Computers erzielt werden. Der vorliegende Kurs zeigt, wie CUDA C (Erweiterung der C-Sprache für die GPU-Programmierung) und das entsprechende (sehr flexible!) CUDA-Laufzeit-API-Framework verwendet werden kann, um die Ausführung einiger typischer Programmiermuster um einen Faktor von 10 oder mehr zu beschleunigen das der CPU. Ausgehend vom CUDA-Programmiermodell geht man zum CUDA-Ausführungsmodell über und betrachtet grundlegende konzeptionelle, Software- und Hardwareprobleme, die zum Verständnis der Funktionsweise von GPUs beitragen. Fallstudien zu mehreren Problemen mit massiv parallelen Algorithmen, die in GPUs implementiert sind, werden ebenfalls weiter ausgeführt. Das theoretische Wissen, das in den Vorlesungen vermittelt wird, wird durch eine Vielzahl von praktischen Beispielen untermauert, an denen die SchülerInnen zu Hause arbeiten können.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Hausarbeit					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestehen der Hausarbeit					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

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: Grundlagen der FEM</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Jedes Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Grundlagen der FEM (129015)			<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. Daniel Balzani Lehrende: Prof. Dr. Daniel Balzani					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  B.Sc. Maschinenbau					
<b>Vorkenntnisse</b> Mathematik im Bachelor-Studium					
<b>Lernziele (learning outcomes)</b>  Die Studierenden <ul style="list-style-type: none"> <li>• erwerben Kenntnisse der Finiten-Elemente Methode und sind in der Lage geeignete Elementtypen sowie Diskretisierungen für unterschiedliche Problemstellungen zu identifizieren bzw. zu konstruieren</li> <li>• können Software kompetent für klassische Probleme des Ingenieurbaus zum Einsatz bringen</li> <li>• werden in die Lage versetzt, eigene Elementroutinen zu programmieren bzw. in einem Software Framework zu implementieren</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Einführung in die Tensorrechnung und Kontinuumsmechanik kleiner Verzerrungen</li> <li>• Variationsprinzipien</li> <li>• Galerkin-Verfahren</li> <li>• Methodik der finiten Elemente</li> <li>• Dreidimensionale Stabwerkselemente und Assemblierungsoperator</li> <li>• Darstellung in Flächen- und Volumenkoordinaten</li> <li>• Isoparametrisches Konzept</li> <li>• Numerische Integration</li> <li>• Weitergehende Elementtypen wie Balken und Platten</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Klausur (120 Minuten) Mündliche Prüfung (15-45 Minuten) Die Prüfungsform wird je nach Teilnehmerzahl am Anfang eines jeden Semesters festgelegt.					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung schriftlich oder mündlich.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 5/105: M.Sc. Angewandte Informatik					

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

<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> see examination regulations/ siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Highlights of Theoretical Computer Science (211057)			<b>Kontaktzeit</b> 6 SWS (90 h)	<b>Selbststudium</b>	<b>Gruppengröße</b> 30 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Successful completion of an introductory course on theoretical computer science (covering formal languages, basics of complexity theory including NP-completeness and reductions, basics of computability theory). Interest and motivation to learn about theoretical concepts.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Michael Walter Prof. Dr. Thomas Zeume Lehrende: Prof. Dr. Michael Walter Prof. Dr. Thomas Zeume Dr. Vladimir Lysikov					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit / Informationstechnik  M.Sc. IT-Sicherheit / Netze und Systeme					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> You will know some of the most important results and insights of modern theoretical computer science. You will learn approaches and techniques that go well beyond a first course. You will be able to recognize when these can be used and how to adapt them to new situations. You will be able to independently acquire new knowledge in this area.					
<b>Inhalt</b> The insights and techniques of modern theoretical computer science have been key for advances in all areas of computer science. In this course, we will discuss some highlights and the techniques that underpin them.  Possible topics that we might cover: <ul style="list-style-type: none"> <li>• Computational models (is there life beyond Turing machines?)</li> <li>• Kolmogorov complexity (what is the shortest program that produces some output?)</li> <li>• Communication complexity (how many bits must Alice and Bob exchange to jointly solve a problem?)</li> <li>• Fine-grained complexity (are some easy problems easier than others? and why?)</li> <li>• Fast multiplication of numbers and matrices (can you beat the high-school method?)</li> <li>• Randomness (does it really help to compute faster?)</li> <li>• Circuit lower bounds (why is it so hard to prove that problems are hard?)</li> <li>• Convex optimization (how to maximize profit if all you can ask are yes/no questions)</li> <li>• Hardness of approximation (how easy is it to find near-optimal solutions?)</li> </ul>					

- Cryptography and computation

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

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

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

**Voraussetzungen für die Vergabe von Credits**

Bestandene Modulabschlussprüfung (Klausur 180 Minuten oder mündliche Prüfung 15-45 Minuten)

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

9/97: M.Sc. Computer Science

9/105: M.Sc. Angewandte Informatik

9/91: M.Sc. IT-Sicherheit / Informationstechnik [PO 22]

9/84: M.Sc. IT-Sicherheit / Informationstechnik [PO 20]

9/99: M.Sc. IT-Sicherheit / Netze und Systeme [PO 22]

9/96: M.Sc. IT-Sicherheit / Netze und Systeme [PO 20]

**Titel des Moduls: High-Performance Computing on Clusters**  
**High-Performance Computing on Clusters**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> High-Performance Computing on Clusters (127511)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Andreas Vogel Lehrende: Prof. Dr. Andreas Vogel					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. Bauingenieurwesen  M.Sc. Subsurface Engineering					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> After successfully completing the module, the students <ul style="list-style-type: none"> <li>• are enabled to design and create programs for parallel computing clusters,</li> <li>• can critically evaluate distributed-memory systems and programming patterns,</li> <li>• can assess the mathematical properties of iterative solvers and their scalability.</li> </ul>					
<b>Inhalt</b> The lecture deals with the parallelization on cluster computers. Distributed-memory programming concepts (MPI) are introduced and best-practice implementation is presented based on applications from scientific computing including the finite element method and machine learning.  Special attention is paid to scalable solvers for systems of equations on distributed-memory systems, focusing on iterative schemes such as simple splitting methods (Richardson, Jacobi, Gauß-Seidel, SOR), Krylov-methods (Gradient descent, CG, BiCGStab) and, in particular, the multigrid method. The mathematical foundations for iterative solvers are reviewed, suitable object-oriented interface structures are developed and an implementation of these solvers for modern parallel computer architectures is developed.  Numerical experiments and self-developed software implementations are used to discuss and illustrate the theoretical results.					
<b>Lehrformen</b> Beamer, Computerlabor, numerische Experimente					
<b>Prüfungsformen</b> Klausur (120 Minuten)					

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: High-Performance Computing on Multicore Processors</b> High-Performance Computing on Multicore Processors					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> High-Performance Computing on Multicore Processors (126509)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Andreas Vogel Lehrende: Prof. Dr. Andreas Vogel					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. Bauingenieurwesen  M.Sc. Subsurface Engineering					
<b>Vorkenntnisse</b> Keine					
<b>Lernziele (learning outcomes)</b> After successfully completing the module, the students <ul style="list-style-type: none"> <li>• are enabled to design and create programs for multicore processors,</li> <li>• can critically evaluate multi-threaded programs and shared-memory access patterns,</li> </ul> can assess the benefits and challenges of multicore programming techniques.					
<b>Inhalt</b> The lecture addresses parallelization on multicore processors. Thread-based programming concepts and techniques, including pthreads, C++11 threads, OpenMP and SYCL, are introduced and best practices are highlighted using applications from scientific computing.  An overview of the relevant hardware aspects including multicore architectures and memory hierarchies is provided. An in-depth introduction to multi-threaded programming on multicore systems with special emphasis on shared-memory parallelization is given and parallelization patterns, thread management and memory access strategies are discussed.  In hands-on sessions, programming exercises are used to discuss and illustrate the presented content.					
<b>Lehrformen</b>  Vorlesung mit Übung					
<b>Prüfungsformen</b> Schriftliche Klausur (120 Minuten)					

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Introduction to Neural Data Science</b> Introduction to Neural Data Science					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Introduction to Neural Data Science (212014)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Robert Schmidt Lehrende: Prof. Dr. Robert Schmidt					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Basic knowledge of calculus and linear algebra, programming in Python.					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"> <li>• Know what signals can be measured from brain activity and how they are processed using data science methods</li> <li>• Be aware of challenges in neuroscience data sets and how they can be addressed using machine learning methods</li> <li>• Apply data analysis methods to neural data and visualize and interpret the results</li> </ul>					
<b>Inhalt</b> Rapid technological advances have recently opened up new possibilities in understanding how the brain works. In particular the number of neurons that can be simultaneously recorded has increased considerably to hundreds (and soon thousands!) of neurons. However, this has led to a big challenge on how to actually process and analyze the resulting big data sets. Solutions for these challenges are part of the new exciting research field of 'Neural Data Science'. In this module you will learn how methods and approaches from data science and machine learning can be applied to study brain signals and the related cognitive functions. In the first part of the module we will focus on so-called spike trains, how they can be analyzed, visualized, and decoded. In the second part of the module we will look at continuous signals, in particular at neural oscillations. Finally, we will learn about and apply some advanced methods from machine learning, such as dimensionality reduction approaches, reinforcement learning, clustering, and computational statistics. In the lectures I will provide the relevant neurobiological background and explain the computational approaches, which will then be applied in the computer exercises using real neural data sets.					
<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 120 ECTS)</b> 6/105: M.Sc. Angewandte Informatik					

<b>Titel des Moduls: Kernanwendungen der CL (kein Angebot im WS 25/26)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 30 h (2 SWS)	<b>Selbststudium</b> 210 h	<b>Gruppengröße</b> 5 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Stefanie Dipper Lehrende: Prof. Dr. Stefanie Dipper, Dr. Ronja Laarmann-Quante					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Grundlegendes Wissen über die zentralen Konzepte, Modelle und Methoden der Computerlinguistik und fortgeschrittene Programmierfähigkeiten in Python.					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls verfügen die Studierenden über vertieftes Wissen und Fertigkeiten in ausgewählten Teilgebieten der Computerlinguistik. <ul style="list-style-type: none"> <li>• Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung. Sie kennen jeweils verschiedene Ansätze und Methoden, die in diesen Bereichen Anwendung finden, und können diese miteinander vergleichen und nach ihren Stärken und Schwächen bewerten.</li> <li>• Sie können die passende Methode für eine Aufgabe auswählen und sie implementieren und evaluieren. Zudem können sie zu ihrer Implementation eine entsprechende wissenschaftliche Publikation verfassen.</li> </ul>					
<b>Inhalt</b> In den im Modul angebotenen Seminaren werden beispielhaft Themen aus dem Kernspektrum der Computerlinguistik abgedeckt, wie z. B. automatische Textzusammenfassung, Koreferenzauflösung. In jedem Themenbereich arbeiten die Studierenden eigenständig mit computerlinguistischer Fachliteratur, ordnen diese ein und bewerten sie und implementieren ausgewählte Ansätze nach.					
<b>Lehrformen</b> <ul style="list-style-type: none"> <li>• Seminaristischer Unterricht.</li> <li>• Gruppenarbeiten.</li> <li>• Mündliche Präsentationen von Forschungsarbeiten.</li> </ul>					
<b>Prüfungsformen</b> Implementation und Hausarbeit.					
<b>Voraussetzungen für die Vergabe von Credits</b>  Erfolgreicher Abschluss der Modulprüfung sowie bestandene Studienleistung in, die seminarspezifisch festgelegt wird.  In der Lehrveranstaltung steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht im Allgemeinen Anwesenheitspflicht					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 8/105: M.Sc. Angewandte Informatik					

## **Titel des Moduls: Klbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Klbox: Lösung von Nachhaltigkeits-Challenges durch den Einsatz künstlicher Intelligenz			<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. Christian Meske Lehrende: Prof. Dr. Jens Pöppelbuß Prof. Dr. Christian Meske					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Es werden keinerlei Vorkenntnisse benötigt, aber Interesse an Methoden der künstlichen Intelligenz. Erste Erfahrungen mit Python sind hilfreich und gern gesehen, jedoch keine Voraussetzung.					
<b>Lernziele (learning outcomes)</b> Die Studierenden erwerben durch das Absolvieren der Lehrveranstaltung folgende Kompetenzen: <ul style="list-style-type: none"> <li>• Die „Kickbox“-Innovationsmethode erläutern und anwenden,</li> <li>• Beispielhafte Nachhaltigkeits-Challenges identifizieren und analysieren,</li> <li>• Potenziale von KI zur Lösung von Nachhaltigkeits-Challenges identifizieren,</li> <li>• Kreativitätstechniken, Ideation- und Prototyping-Methoden (z. B. aus dem Design Thinking sowie Lean-Start-Up) sowie Nudging-Elemente für den eigenen Innovationsprozess auswählen und anwenden,</li> <li>• Eigene kreative Ideen präsentieren und testen und mittels Feedbacks überarbeiten,</li> <li>• Ziele definieren, Projekte managen und Meilensteine eigenständig erreichen, Teamarbeit unter Einsatz von begrenzten Ressourcen koordinieren,</li> <li>• KI-Lösungen prototypisch implementieren,</li> <li>• Unternehmerisches Handeln,</li> <li>• Nachhaltigkeit anhand etablierter Dimensionen (z. B. Triple-Bottom-Line und Corporate Social Responsibility) zur Bewertung eigener Ideen nutzen.</li> </ul>					
<b>Inhalt</b> <ul style="list-style-type: none"> <li>• Kickbox-Innovationsmethode</li> <li>• Nachhaltigkeit, Nachhaltigkeitsziele und -herausforderungen (z.B. E-Waste-Reduktion im Gesundheitssektor)</li> <li>• Künstliche Intelligenz sowie Nudging-Elemente</li> </ul>					
<b>Lehrformen</b> <ul style="list-style-type: none"> <li>• Vereinzelt Vorlesungen/ Inputs von Dozierenden</li> <li>• Projektarbeit und Projektveranstaltungen als Schwerpunkt</li> <li>• Feedback und Möglichkeiten zum Austausch mit Dozierenden</li> </ul>					
<b>Prüfungsformen</b> Es sind zwei Prüfungsleistungen zu erbringen. Abschlusspräsentation (Gruppenleistung, Dauer: 15 Minuten plus anschließende Diskussion von 10 Minuten; 100 % der Endnote) Abschlussbericht (Gruppenleistung; Bestehen notwendig zum Abschluss des Moduls und zum Erwerb der					

Leistungspunkte)

Prüfungstermine: Die Prüfungstermine werden zu Beginn des Semesters bekanntgegeben.

**Voraussetzungen für die Vergabe von Credits**

- Prüfungsleistung bestanden und
- Um das Modul erfolgreich abzuschließen ist es außerdem notwendig, an mindestens 60% der Treffen im Plenum teilzunehmen

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Kryptographie</b> Cryptography					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Kryptographie (212017)			<b>Kontaktzeit</b> 6 SWS (90 h)	<b>Selbststudium</b> 150 h	<b>Gruppengröße</b> 100 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Alex May Lehrende: Prof. Dr. Alex May					
<b>Verwendung des Moduls</b> B.Sc. IT-Sicherheit/ Informationstechnik  M.Sc. IT-Sicherheit/ Netze und Systeme  M.Sc. Computer Science  M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Inhalte der Vorlesungen Einführung in die Kryptographie 1 und 2					
<b>Lernziele (learning outcomes)</b>  Die Studierenden haben ein Verständnis der wesentlichen mathematischen Methoden und Verfahren, auf denen moderne kryptographische Verfahren beruhen. Die Tiefe der Behandlung der Verfahren geht deutlich über das in den vorhergehenden Veranstaltungen vermittelte Maß hinaus. Die Teilnehmer sind zur Analyse und dem Design aktueller und zukünftiger kryptographischer Methoden befähigt. Zudem weisen sie ein Bewusstsein für Methodik und Mächtigkeit verschiedenster Angriffsszenarien auf.					
<b>Inhalt</b> Es wird eine Einführung in moderne Methoden der symmetrischen und asymmetrischen Kryptographie geboten. Dazu wird ein Angreifermodell definiert und die Sicherheit der vorgestellten Verschlüsselungs-, Hash- und Signaturverfahren unter wohldefinierten Komplexitätsmaßnahmen in diesem Angreifermodell nachgewiesen.  Themenübersicht:  <ul style="list-style-type: none"> <li>• Sichere Verschlüsselung gegenüber KPA-, CPA- und CCA-Angreifern</li> <li>• Pseudozufallsfunktionen und -permutationen</li> <li>• Message Authentication Codes</li> <li>• Kollisionsresistente Hashfunktionen</li> <li>• Blockchiffren</li> <li>• Konstruktion von Zufallszahlengeneratoren</li> <li>• Diffie-Hellman Schlüsselaustausch</li> <li>• Trapdoor Einwegpermutationen</li> <li>• Public Key Verschlüsselung: RSA, ElGamal, Goldwasser-Micali, Rabin, Paillier</li> <li>• Einwegsignaturen</li> <li>• Signaturen aus kollisionsresistenten Hashfunktionen</li> </ul>					

- Random-Oracle Modell

**Lehrformen**

Vorlesung und Übungen

**Prüfungsformen**

Klausur (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

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

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

8/99: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 22]

8/96: M.Sc. IT-Sicherheit/ Netze und Systeme [PO 20]

8/97: M.Sc. Computer Science

8/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Machine Learning: Supervised Methods (kein Angebot im SS 25)**  
**Machine Learning: Supervised Methods (no offer in SS 25)**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Machine Learning: Supervised Methods (211024)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 80 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/ Informationstechnik					
<b>Vorkenntnisse</b> empfohlen: Vorlesung "Mathematics for Modeling and Data Analysis"					
<b>Lernziele (learning outcomes)</b> Internationalisierung: Die Veranstaltung wird auf Englisch durchgeführt. Digitalisierung: Inhalte werden durch Videos und Lesematerial vermittelt. Übungsaufgaben mit Programmieranteilen werden in Form von Jupyter-Notebooks bereitgestellt. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• verstehen die Teilnehmer die Grundlagen der statistischen Lerntheorie</li> <li>• kennen die Teilnehmer die wichtigsten Algorithmen des überwachten statistischen Lernens und können diese auf Lernprobleme anwenden,</li> <li>• kennen die Teilnehmer Stärken und Beschränkungen verschiedenen Lernverfahren,</li> <li>• können die Teilnehmer Standardsoftware zum maschinellen Lernen zur Lösung neuer Probleme einsetzen.</li> </ul>					
<b>Inhalt</b> Grundlagen der statistischen Lerntheorie, Querschnitt der wichtigsten Algorithmen des maschinellen Lernens, konkrete Problemlösung mit Standardsoftware.					
<b>Lehrformen</b> Vorlesung mit Übung im flipped classroom Format					
<b>Prüfungsformen</b> Klausur (90 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Das Bestehen des Kurses ist ein zweistufiger Prozess. Die erste Stufe ist ein aktiver Beitrag während des Semesters, dessen Einzelheiten in einer in einer der ersten Sitzungen bekannt gegeben werden. Die zweite Stufe ist eine Klausur von 90 Minuten. Der aktive Beitrag während des Semesters wird nicht benotet, ist aber Voraussetzung für die Teilnahme an der Prüfung. Die Note wird basiert ausschließlich auf der Abschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>					

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

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

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

**Titel des Moduls: Machine Learning: Unsupervised Methods**  
**Machine Learning: Unsupervised Methods**

<b>Modul-Nr./Code</b>	<b>Credits</b> 10 CP	<b>Workload</b> 300 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Machine Learning: Unsupervised Methods (212501)			<b>Kontaktzeit</b> 8 SWS (120h)	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> 40 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Laurenz Wiskott Lehrende: Prof. Dr. Laurenz Wiskott					
<b>Verwendung des Moduls</b> M.Sc. Computer Science  M.Sc. Angewandte Informatik  M.Sc. Elektro-und Informationstechnik					
<b>Vorkenntnisse</b> Lineare Algebra (Vektoren, Matrizen, Eigenvektoren, Eigenwerte, ...), Analysis (Funktionen, Ableitungen, ...), Wahrscheinlichkeitstheorie (joint/marginal/contidional probabilities in multiple variabels, Bayesian theorem, ...), Programmierung (fließend in mindestens einer Programmiersprache, idealerweise Python).					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss dieses Kurses können die Teilnehmer: <ul style="list-style-type: none"> <li>• kennen die Grundlagen des maschinellen Lernens,</li> <li>• kennen sie eine Reihe von wichtigen unüberwachten Lernmethoden,</li> <li>• können diskutieren und entscheiden, welche der Methoden für einen gegebenen Datensatz geeignet sind,</li> <li>• verstehen die Mathematik dieser Methoden,</li> <li>• wissen, wie man diese Methoden in Python implementiert und anwendet,</li> <li>• haben Erfahrung in der Organisation und Arbeit in einem Team gesammelt,</li> <li>• kennen Problemlösungsstrategien wie Brainstorming,</li> <li>• kann über all dies auf Englisch kommunizieren.</li> </ul>					
<b>Inhalt</b> Dieser Kurs führt zunächst in das Gebiet des maschinellen Lernens ein und behandelt einige grundlegende Konzepte wie Lernparadigmen, Training, Testen und Verallgemeinerung sowie Over-/Underfitting. Anschließend werden verschiedene unüberwachte Methoden des klassischen maschinellen Lernens behandelt, wie z. B. Hauptkomponentenanalyse, unabhängige Komponentenanalyse, Vektorquantisierung, Clustering, Bayes'sche Theorie und grafische Modelle. Dies sind oberflächliche Methoden, dieser Kurs deckt standardmäßig keine Deep-Learning-Methoden ab, aber die Studenten können Deep-Learning in den Problemen verwenden. Auf der praktischen Seite erhalten die Studierenden einfache Probleme, die sie in Gruppenarbeit lösen und dann dem gesamten Kurs präsentieren müssen. In diesem Zusammenhang wird der Kurs auch einige Soft Skills wie Concept Mapping, Teamreflexion und Brainstorming behandeln.					
<b>Lehrformen</b> Vorlesung + Übungen zum Selbststudium + Diskussion der Vorlesung und der Übungen im umgekehrten Unterrichtsstil + praktische Problemlösung in Gruppen + gegenseitige Überprüfung + Präsentation der Ergebnisse.					
<b>Prüfungsformen</b> Bestehen/Nichtbestehen bei Gruppenarbeit plus benotete Klausur von 90 Minuten.					

**Voraussetzungen für die Vergabe von Credits**

Bestehen der Gruppenarbeit und der Klausur.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

10/105: M.Sc. Angewandte Informatik

10/97: M.Sc. Computer Science

**Titel des Moduls: Master-Praktikum Deep Learning and Natural Language Processing  
(kein Angebot im SS 25)**

<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> 30 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 3 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> After successful completion of this module, <ul style="list-style-type: none"> <li>• The students should be able to develop applications using deep learning techniques for machine learning tasks.</li> <li>• The students should be able to measure the efficacy of the developed machine learning applications using scientific methods.</li> <li>• The students should be familiar with PyTorch, a library for implementing neural network based machine learning applications and experiments</li> </ul>					
<b>Inhalt</b> This praktikum teaches the basics of the development of applications for machine learning tasks. The praktikum is aimed at but not necessarily limited to natural language processing (NLP) tasks, such as machine translation or sentiment analysis. NLP tasks are aimed at the ability to process or generate natural language (e.g. English) texts. The course is started with a short introduction to deep learning, NLP, Python programming and a development environment setup (installing dependencies and setting up an IDE (e.g. PyCharm)). Subsequently, teams of up to 3 students will choose (and may propose) topics that consist of a machine learning task. These tasks have to be solved by the students over the course of one semester, with regular meetings to discuss progress.					
<b>Lehrformen</b> Introductory lectures in the first weeks, followed by independent work on the chosen practical task, supported by regular meetings.					
<b>Prüfungsformen</b> Written report describing the motivation, methodology and result of the chosen task in Latex (max. 10 pages in ACL format), code for the task and its documentation, final presentation using slides (15 min. + 5 min. discussion)					
<b>Voraussetzungen für die Vergabe von Credits</b> Written report, documented code, participation in meetings, final presentation					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 5/105: M.Sc. Angewandte Informatik					

**Titel des Moduls: Master-Praktikum: Autonomous Robotics [letztmalig SS 25]**

<b>Modul-Nr./Code</b>	<b>Credits</b> 3 CP	<b>Workload</b> 90 h	<b>Semester</b>	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Autonomous Robotics			<b>Kontaktzeit</b> 40h Blockpraktikum	<b>Selbststudium</b> 50 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Greogor Schöner Lehrende: Prof. Dr. Greogor Schöner					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Vorlesung Autonomous Robotics: Action, Perception, and Cognition					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"><li>• Einüben des praktischen Arbeitens im Umfeld der autonomen Robotik, insbesondere der Entwurf von mathematischen Verfahren und ihre algorithmische Implementation zur Erzeugung von zeitlich strukturierten Abläufen</li><li>• Einüben der Dokumentation von technischen Programmieraufgaben.</li><li>• Verständnis der Grundlagen der autonomen Robotik anhand des einfachsten Robotersystems, eines autonomen Kleinstvehikels.</li><li>• Einüben des Lesens und Schreibens von wissenschaftlich-technischen Artikeln</li></ul>					
<b>Inhalt</b> <p>Ziel des Praktikums ist es, die Grundlagen der autonomen Robotik anhand von Experimenten zu vermitteln, die eine enge Kopplung an aktuelle wissenschaftliche Probleme dieses Feldes aufweisen. Am Beispiel der Kleinstroboter vom Typ E-Puck wird hierbei die Erzeugung und Organisation von künstlichem Verhalten behandelt. Unter Verwendung der robotereigenen Sensorik werden experimentelle Aufgaben durch Implementierungen in der Programmierumgebung MATLAB gelöst. Der theoretische Hintergrund der Lösungen ist der dynamische Systeme Ansatz der kognitiven Robotik.</p> <p>Eine optionale Erweiterung des Praktikums durch das der dritte CP erzielt wird, ist das Lesen einer wissenschaftlichen Veröffentlichung im Umfeld der im Praktikum behandelten Methoden und das Schreiben eines Essays, in dem Fragen zu dem Artikel durch einen längeren, strukturierten und illustrierten Text beantwortet werden.</p>					
<b>Lehrformen</b> Praktikum					
<b>Prüfungsformen</b> Praktische Prüfung					
<b>Voraussetzungen für die Vergabe von Credits</b> Erfolgreiche Bearbeitung der Programmieraufgaben; Abgabe von schriftlichen Dokumentationen					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 3/105: M.Sc. Angewandte Informatik					

<b>Titel des Moduls: Master-Praktikum: Big Data in der Bioinformatik</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> 1 Semester
<b>Lehrveranstaltungen</b> Big Data in der Bioinformatik (202621 bzw. 190801)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Axel Mosig Lehrende: Prof. Dr. Axel Mosig					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> praktische Programmiererfahrung in einer oder mehreren Sprachen (z. B. Java, R, Python, Matlab) Vertiefungsmodul(e) aus dem Themengebiet der Bioinformatik					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• können Studierende praktischen Herausforderungen bei der Entwicklung von Bioinformatik-Anwendungen begegnen,</li> <li>• haben die Studierenden Programmier-Bibliotheken aus dem Bereich Bioinformatik kennen gelernt</li> <li>• haben Studierenden die Verwendung von Workflow-Systemen eingeübt</li> <li>• haben Studierende die Mechanismen von Code- und Programm-Dokumentation eingeübt</li> <li>• haben Studierende Verfahren zur Bereitstellung eigener Libraries (z. B. R-Pakete, .jar-Files) angewendet</li> </ul>					
<b>Inhalt</b> Die Bioinformatik wendet naturgemäß Informatik-Methoden auf die Daten eines lebenswissenschaftlichen Anwendungsfaches an, stellt also per se angewandte Digitalisierung dar. Das Praktikum vermittelt Grundlagen der Programmierung mit Bezug zu lebenswissenschaftlichen Anwendungen mit großen Datenmengen. Dies geschieht anhand aktueller Beispiele aus den Themengebieten Bildverarbeitung und Sequenzanalyse. Nach einer kurzen Einführung (Präsenztreffen) in Programmierung und Entwicklungsumgebungen (z. B. Java, R, C++, eclipse, RStudio, Python, Matlab) werden praktische Programmieraufgaben ausgegeben und im Laufe des Semesters bearbeitet, ggfls. mit weiteren Präsenztreffen zur Diskussion des Fortschritts.					
<b>Lehrformen</b> Einführung als seminaristischer Unterricht, Bearbeitung der praktischen Aufgabe selbständig oder als Gruppenarbeit.					
<b>Prüfungsformen</b> Semesterbegleitend; Praktische Prüfung: Protokoll max. 10 Seiten (wenn Abgabe bis zwei Wochen nach Praktikumsende, dann Korrekturrunde vor Benotung), Dokumentation der praktischen Aufgabe					
<b>Voraussetzungen für die Vergabe von Credits</b> Protokoll (siehe Prüfungsformen), Bearbeitung / Dokumentation der praktischen Aufgabe, Teilnahme an den Präsenztreffen, Abschlusspräsentation (15 min. + 5 min. Diskussion)					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 5/105: M.Sc. Angewandte Informatik					

**Titel des Moduls: Master-Praktikum: Computational Proteomics**  
**Master-Practical „Computational Proteomics“**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Master-Praktikum: Computational Proteomics (201914)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 3 Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Martin Eisenacher  
 Lehrende: Prof. Dr. Martin Eisenacher  
 Prof. Dr. Julian Uszkoreit

**Verwendung des Moduls**

M.Sc. Angewandte Informatik

**Vorkenntnisse**

praktische Programmiererfahrung in einer oder mehreren Sprachen (z. B. Java, R, Python, Matlab)  
 Vertiefungsmodul(e) aus dem Themengebiet der Bioinformatik (z.B. Vorlesung "Bioinformatics for Proteomics")

**Lernziele (learning outcomes)**

Nach erfolgreichem Abschluss des Moduls können die Studierenden:

- sich bei der Entwicklung von Bioinformatik-Anwendungen an praktischen Herausforderungen beteiligen,
- haben sie Programmierbibliotheken aus dem Bereich der Bioinformatik kennengelernt,
- haben den Einsatz von Workflow-Systemen geübt,
- haben die Mechanismen der Code- und Programmdokumentation erlernt,
- haben Verfahren zur Erstellung eigener Bibliotheken (z. B. Python-Module, R-Pakete, Rust-Kisten) eingeübt

Fächerspezifische Kompetenzen:

- Programmierung von Lösungen für bioinformatische Anwendungen
- Umgang mit modernsten Werkzeugen / Formaten und Programmierbibliotheken der Bioinformatik
- Umgang mit großen Datenmengen

Übergreifende Kompetenzen:

- Instrumentelle Kompetenzen ("Instrumente nutzen")
- Projektmanagement / Organisation und Software-Engineering mit digitalen Werkzeugen
- Wissenschaftlich strukturierte Dokumentation
- Systemische Kompetenzen ("wissen, wie die Dinge im wirklichen Leben funktionieren")
- Teamarbeit und Teamfähigkeit
- Selbstständiges Lernen und Arbeiten
- Kommunikative Kompetenzen ("ziel- und anlassbezogen angemessen kommunizieren")
- Präsentation von wissenschaftlichen Ergebnissen
- Verwendung englischer Fachbegriffe aus der Bioinformatik
- Rhetorische und sprachliche Kompetenz (Deutsch / Englisch)

**Inhalt**

Die Bioinformatik wendet naturgemäß Methoden der Informatik auf die Daten eines lebenswissenschaftlichen Anwendungsgebietes an, d. h. sie stellt die angewandte Digitalisierung schlechthin dar. Das Praktikum vermittelt die Grundlagen der Programmierung/Entwicklung mit Bezug zu Life Science Anwendungen unter Verwendung von Daten der Hochdurchsatz-Massenspektrometrie/Proteomics/und anderer Omics.

Nach einer kurzen Einführung (Präsenzveranstaltung) in Programmier- und Entwicklungsumgebungen (z.B.

Python, R, Rust, RStudio, Visual Studio Code, KNIME, nextflow, github) werden im Laufe des Semesters praktische Programmieraufgaben aufgenommen und bearbeitet, ggf. mit weiteren Präsenzveranstaltungen zur Besprechung der Fortschritte.

**Lehrformen**

Einführungswochen, selbständige Bearbeitung der praktischen Aufgabe allein oder in Gruppenarbeit.

**Prüfungsformen**

Abschließende Folienpräsentation (20 Min. + 10 Min. Diskussion)

**Voraussetzungen für die Vergabe von Credits**

Bearbeitung / Dokumentation der Praxisaufgabe, Teilnahme an den Präsenzveranstaltungen, Abschlusspräsentation

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Mobile Network Security</b> Mobile Network Security					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Mobile Network Security (211012)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> English			<b>Teilnahmevoraussetzungen</b> None		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Katharina Kohls Lehrende: Prof. Dr. Katharina Kohls					
<b>Verwendung des Moduls</b>					
<b>Vorkenntnisse</b> The lecture focuses on mobile network security. Prior knowledge in the context of computer networks and their security mechanisms helps to understand the technical concepts that will be addressed in the lecture.					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"> <li>• Knowing mobile network architectures and their components in 4G and 5G networks</li> <li>• Understanding existing attacks and their attack vectors, as well as security mechanisms that avoid known attacks</li> <li>• Experience with scientific work in mobile network security</li> </ul>					
<b>Inhalt</b> Mobile networks are an integral part of our everyday lives. Their use cases range from casual web browsing over campus networks in industrial environments to first responder communication. In this course, we cover the technical aspects of mobile networks and address their security capabilities. After an introduction to the technical foundations of mobile network deployments, we will go into detail with scientific work on existing attacks against 4G and 5G networks. To this end, we analyze open attack vectors and discuss the consequences of attacks if being conducted in real-world infrastructures.					
<b>Lehrformen</b> The course consists of lectures that provide theoretical knowledge and practical exercises that help to apply the contents of the lectures.					
<b>Prüfungsformen</b> Written exam with a duration of 120 minutes.					
<b>Voraussetzungen für die Vergabe von Credits</b> Passing the exam					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>  5/97: M.Sc. Computer Science  5/105: M.Sc. Angewandte Informatik  5/91: M.Sc. IT-Sicherheit / Informationstechnik  5/99: M.Sc. IT-Sicherheit/ Netze und Systeme					

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

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Natural language processing with deep learning (212038)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Eigenes Laptop mit Python und einer geeigneten IDE für die Übungen mitbringen		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Ivan Habernal Lehrende: Prof. Dr. Ivan Habernal					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b> Praktisch: Kenntnisse in Python (Projekt-Strukturen, OOP, Unit-Tests), Kenntnisse in Numpy sind von Vorteil Theoretisch: Grundlagen der Analysis (Ableitungen) und der linearen Algebra (Vektoren), obwohl wir das wiederholen werden. Einige Grundkenntnisse der Linguistik (z. B. Syntax) auf Oberschulniveau sind ebenfalls von Vorteil.					
<b>Lernziele (learning outcomes)</b> Die Studierenden <ul style="list-style-type: none"><li>• verstehen das Rückgrat des modernen NLP, wie z.B. Embeddings und Transformer-basierte Modelle</li><li>• können die Fähigkeiten und Schwächen verschiedener Deep-Learning-Modelle im NLP kritisch beurteilen</li><li>• verstehen verschiedene Tasks im NLP, deren Bewertung und Modellierungsannahmen</li><li>• können verschiedene Modelle und Ansätze in Python implementieren und sammeln praktische Erfahrungen</li><li>• verstehen fortgeschrittene moderne Paradigmen in Large Language Models</li></ul>					
<b>Inhalt</b> Typische Aufgaben und Datensätze der natürlichen Sprachverarbeitung (NLP) und ihre Bewertung. Warum ist NLP schwierig? Auffrischung der mathematischen Grundlagen, Calculus, gradientenbasierte Optimierung, Backpropagation für beliebige Funktionen. Log-lineare Modelle und Textklassifikation. Tiefe neuronale Netze. Sprachmodelle und Word embeddings. Lernen statischer Word embeddings. Rekurrente neuronale Netze. Encoder-Decoder, Text Generation, Attention und autoregressive Modelle. Transformers. Self-Attention und BERT. Reine Decoder-Modelle und GPT. LLMs: Prompting und In-context Lernen					
<b>Lehrformen</b> Wir werden Vorlesungen und Übungen haben. Die Vorlesungen werden ziemlich interaktiv sein, da versucht wird, die Studierenden in Fragen und kritisches Denken einzubeziehen. Die Folien für jede Vorlesung werden im Voraus hochgeladen, damit die Studierenden sie für ihre Notizen verwenden können. Jede Vorlesung enthält Links zu relevanten Forschungsarbeiten für diejenigen, die sich eingehender mit dem Thema befassen möchten. Die Vorlesungen sind überwiegend theoretisch, d. h. wir behandeln die wichtigsten Konzepte, Ideen und mathematischen Beschreibungen, aber nicht, wie man es in einem bestimmten Framework programmiert. In den praktischen Kursen werden wir genau das Gegenteil tun. Die Studierenden werden mit aktuellen Mainstream-Frameworks für Deep Learning in NLP, wie Pytorch oder Huggingface, experimentieren. Wir werden ein breites Spektrum an Komplexitäten abdecken, von der Programmierung eines einfachen neuronalen Netzes von Grund auf bis hin zur Verwendung eines vortrainierten Sprachmodells.					

**Prüfungsformen**

Klausur (90 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

5/97: M.Sc. Computer Science

**Titel des Moduls: Netzsicherheit 1**  
**Network Security 1**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> 3. Semester	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Netzsicherheit 1 (212012)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Jörg Schwenk  
 Lehrende: Prof. Dr. Jörg Schwenk

**Verwendung des Moduls**

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

**Vorkenntnisse**

Grundkenntnisse in TCP/IP, Grundkenntnisse der Sicherheitsprobleme von Computernetzen auf dem Niveau populärer Fachzeitschriften (z.B. c't).

**Lernziele (learning outcomes)**

Studierende verfügen nach erfolgreichem Abschluss des Moduls über ein umfassendes Verständnis der technischen Aspekte von Netzsicherheit. Sie haben erkannt, dass Kryptographie alleine nicht ausreicht, um sicherheitstechnische Probleme zu lösen. Sie haben ein umfassendes Systemverständnis für komplexe IT-Systeme erworben. Durch eigenständige Überlegungen zur Verbesserung der Netzsicherheit bereiten sich die Studierenden auf ihre Rolle im Berufsleben vor. Sie können neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Sie können im Gespräch den Nutzen der von ihnen erarbeiteten Lösungen argumentativ begründen. Sie haben verstanden, dass nicht-technische Faktoren wie Fragen der Haftung und der entstehenden Kosten Entscheidungen zur IT-Sicherheit maßgeblich mit beeinflussen.

**Inhalt**

You can find our Moodle course via the Moodle Search!

Wenn Kryptographie in einer technischen Umgebung wie einem Computer-, Daten- oder Telefonnetz eingesetzt wird, hängt die Sicherheit außer von rein kryptographischen Faktoren auch von der technischen Einbettung der Verschlüsselungs- und Signaturalgorithmen ab. Prominente Beispiele (für fehlerhafte Einbettungen) sind EFAIL (efail.de), Angriffe auf die WLAN-Verschlüsselungssysteme WEP und WPA (KRACK) und diverse Angriffe auf TLS (Bleichenbacher, POODLE, DROWN, ROBOT). Das Modul „Netzsicherheit 1“ beschäftigt sich mit konkreten Netzen zur Datenübertragung und beleuchtet diese von allen Seiten auf ihre Sicherheit hin. Es umfasst folgende Teile: Einführung: Internet Einführung: Vertraulichkeit Einführung: Integrität Einführung: Kryptographische Protokolle PPP-Sicherheit (insb. PPTP), EAP-Protokolle WLAN-Sicherheit (WEP, WPA, Wardriving, KRACK) GSM- und UMTS-Mobilfunk (Authentisierung und Verschlüsselung) IPSec (ESP und AH, IKEv1 und v2, Angriffe auf IPSec) Dateiverschlüsselung mit OpenPGP (Datenformat, Efail, Klima-Rosa) E Mail-Verschlüsselung mit S/MIME (SMTP, Datenformat, Efail, POP3, IMAP) Neben den Systemen selbst werden dabei auch publizierte Angriffe auf diese Systeme besprochen; die Studierenden stellen selbst wissenschaftliche Überlegungen zur Verbesserung der Sicherheit an.

**Lehrformen**

Den aktuellen Moodle Kurs finden Sie über die [Moodle Suche](#)

Der Inhalt der Vorlesung wird über Youtube-Videos und Materialien in Moodle zur Verfügung gestellt. Ergänzend dazu gibt es in Präsenz eine Vertiefungsvorlesung. Dort werden keine neuen Themen vorgestellt, sondern die Themen der Online-Materialien werden vertiefend behandelt. Ob eine Aufzeichnung der Präsenzveranstaltung möglich ist, muss noch geklärt werden. Durch diese Mischform aus Online-Materialien und Vertiefung in Präsenz soll die Teilnahme aller Studierenden auch bei möglicherweise erhöhtem Krankenstand im Winter gewährleistet werden.

Im WS 24/25 wird die gesamte Vorlesung online angeboten und kann im Hörsaal verfolgt werden.

### **Prüfungsformen**

Klausur (120 Minuten)

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

Bestandene Klausur

### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

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

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

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

5/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Netzsicherheit 2**  
**Network Security 2**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> Siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Netzsicherheit 2 (211013)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 150 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Jörg Schwenk  
 Lehrende: Prof. Dr. Jörg Schwenk

**Verwendung des Moduls**

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

**Vorkenntnisse**

Grundkenntnisse in TCP/IP, Grundkenntnisse der Sicherheitsprobleme von Computernetzen auf dem Niveau populärer Fachzeitschriften (z.B. c't)

**Lernziele (learning outcomes)**

Studierende verfügen nach erfolgreichem Abschluss des Moduls über ein umfassendes Verständnis der technischen Aspekte von Netzsicherheit. Sie haben erkannt, dass Kryptographie allein nicht ausreicht, um sicherheitstechnische Probleme zu lösen. Sie haben ein umfassendes Systemverständnis für komplexe IT-Systeme erworben. Durch eigenständige Überlegungen zur Verbesserung der Netzsicherheit bereiten sich die Studierenden auf ihre Rolle im Berufsleben vor. Sie können neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Sie können im Gespräch den Nutzen der von ihnen erarbeiteten Lösungen argumentativ begründen. Sie haben verstanden, dass nicht-technische Faktoren wie Fragen der Haftung und der entstehenden Kosten Entscheidungen zur IT-Sicherheit maßgeblich mit beeinflussen.

**Inhalt**

Wenn Kryptographie in einer technischen Umgebung wie einem Computer-, Daten- oder Telefonnetz eingesetzt wird, hängt die Sicherheit außer von rein kryptographischen Faktoren auch von der technischen Einbettung der Verschlüsselungs- und Signaturalgorithmen ab. Prominente Beispiele (für fehlerhafte Einbettungen) sind EFAIL (efail.de), Angriffe auf die WLAN-Verschlüsselungssysteme WEP und WPA (KRACK) und diverse Angriffe auf TLS (Bleichenbacher, POODLE, DROWN, ROBOT). Das Modul „Netzsicherheit“ beschäftigt sich mit konkreten Netzen zur Datenübertragung und beleuchtet diese von allen Seiten auf ihre Sicherheit hin. Es umfasst folgende Teile:

- Sicherheit von HTTP (HTTP Authentication, Secure HTTP, Architektur von SSL/TLS)
- Transport Layer Security (TLS1.2, Versionen SSL 2.0 bis TLS 1.3)
- Angriffe auf SSL und TLS (BEAST, CRIME, POODLE, Lucky13, Bleichenbacher, DROWN, Heartbleed, Invalid Curve)
- Secure Shell - SSH
- das Domain Name System und DNSSEC (faktorisierte Schlüssel)
- Sicherheit von Webanwendungen (HTML, URI, XSS, CSRF, SQLi, SSO)
- XML- und JSON-Sicherheit

Neben den Systemen selbst werden dabei auch publizierte Angriffe auf diese Systeme besprochen; die Studierenden stellen selbst wissenschaftliche Überlegungen zur Verbesserung der Sicherheit an.

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Klausur (120 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

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

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

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

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

5/105: M.Sc. Angewandte Informatik

**Titel des Moduls: NLP für low-resource-Daten (kein Angebot im WS 25/26)**

<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 30 h (2 SWS)	<b>Selbststudium</b> 210 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Stefanie Dipper  
Lehrende: Prof. Dr. Stefanie Dipper, Dr. Ronja Laarmann-Quante

**Verwendung des Moduls**

M.Sc. Angewandte Informatik

**Vorkenntnisse**

Grundlegendes Wissen über die zentralen Konzepte, Modelle und Methoden der Computerlinguistik und fortgeschrittene Programmierfähigkeiten in Python.

**Lernziele (learning outcomes)**

Nach dem erfolgreichen Abschluss des Moduls verfügen die Studierenden über vertieftes Wissen und Fertigkeiten in ausgewählten Teilgebieten der Computerlinguistik.

- Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung. Sie kennen jeweils verschiedene Ansätze und Methoden, die in diesen Bereichen Anwendung finden, und können diese miteinander vergleichen und nach ihren Stärken und Schwächen bewerten.
- Sie können die passende Methode für eine Aufgabe auswählen und sie implementieren und evaluieren. Zudem können sie zu ihrer Implementation eine entsprechende wissenschaftliche Publikation verfassen.

**Inhalt**

In den im Modul angebotenen Seminaren werden Themen der Computerlinguistik abgedeckt, für die es vergleichsweise wenige Ressourcen gibt, wie z.B. semantischer Wandel in historischen Daten, und die daher nur bedingt mit aktuellen Methoden wie Large Language Models bearbeitet werden können. In jedem Themenbereich arbeiten die Studierenden eigenständig mit computerlinguistischer Fachliteratur, ordnen diese ein und bewerten sie und implementieren ausgewählte Ansätze nach.

**Lehrformen**

- Seminaristischer Unterricht.
- Gruppenarbeiten.
- Mündliche Präsentationen von Forschungsarbeiten.

**Prüfungsformen**

Implementation und Hausarbeit.

**Voraussetzungen für die Vergabe von Credits**

Erfolgreicher Abschluss der Modulprüfung sowie bestandene Studienleistung in, die seminarspezifisch festgelegt wird.

In der Lehrveranstaltung steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht im Allgemeinen Anwesenheitspflicht.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**  
8/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: NLP im Bildungssektor</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 8 CP	<b>Workload</b> 240 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 30 h (2 SWS)	<b>Selbststudium</b> 210 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Stefanie Dipper Lehrende: Prof. Dr. Stefanie Dipper, Dr. Ronja Laarmann-Quante					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Grundlegendes Wissen über die zentralen Konzepte, Modelle und Methoden der Computerlinguistik und fortgeschrittene Programmierfähigkeiten in Python.					
<b>Lernziele (learning outcomes)</b> Nach dem erfolgreichen Abschluss des Moduls verfügen die Studierenden über vertieftes Wissen und Fertigkeiten in ausgewählten Teilgebieten der Computerlinguistik. <ul style="list-style-type: none"> <li>• Sie haben einen Überblick über mehrere Bereiche computerlinguistischer Forschung. Sie kennen jeweils verschiedene Ansätze und Methoden, die in diesen Bereichen Anwendung finden, und können diese miteinander vergleichen und nach ihren Stärken und Schwächen bewerten.</li> <li>• Sie können die passende Methode für eine Aufgabe auswählen und sie implementieren und evaluieren. Zudem können sie zu ihrer Implementation eine entsprechende wissenschaftliche Publikation verfassen.</li> </ul>					
<b>Inhalt</b> In den im Modul angebotenen Seminaren werden unterschiedliche Themen aus dem Gebiet NLP im Bildungsbereich abgedeckt, wie z.B. automatische Aufgabenerstellung oder Freitextbewertung. In jedem Themenbereich arbeiten die Studierenden eigenständig mit computerlinguistischer Fachliteratur, ordnen diese ein und bewerten sie und implementieren ausgewählte Ansätze nach.					
<b>Lehrformen</b> <ul style="list-style-type: none"> <li>• Seminaristischer Unterricht.</li> <li>• Gruppenarbeiten.</li> <li>• Mündliche Präsentationen von Forschungsarbeiten.</li> </ul>					
<b>Prüfungsformen</b> Implementation und Hausarbeit.					
<b>Voraussetzungen für die Vergabe von Credits</b> Erfolgreicher Abschluss der Modulprüfung sowie bestandene Studienleistung in, die seminarspezifisch festgelegt wird.  In der Lehrveranstaltung steht die Diskurs-, Handlungs- oder Praxisorientierung im Vordergrund. Daher besteht im Allgemeinen Anwesenheitspflicht.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 8/105: M.Sc. Angewandte Informatik					

<b>Titel des Moduls: Numerical Optimization</b> Numerical Optimization					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Numerical Optimization (212043)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 120h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Für diesen Kurs gibt es keine formalen Anforderungen. Zur Zielgruppe gehören Masterstudenten aller technischen Fächer, wie Informatik, Mathematik, Physik und Ingenieurwesen.		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Tobias Glasmachers Lehrende: Prof. Dr. Tobias Glasmachers					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b> Dieser Kurs erfordert ein solides Verständnis verschiedener mathematischer Konzepte zur Definition und Analyse von Optimierungsalgorithmen mit mathematischen Werkzeugen, einschließlich einiger Beweise. Die Teilnehmer benötigen ein solides Verständnis der linearen Algebra (Matrizen, Eigenwertzerlegung, positive Definitheit), der Analysis (Sequenzen, Konvergenz, Konvexität, Gradient und Hesse Matrix) und der Wahrscheinlichkeitstheorie (multivariate Normalverteilung). Vorkenntnisse in Numerik sind von Vorteil, aber nicht erforderlich. Sowohl für die Übungen als auch für die Prüfung sind Grundkenntnisse der Python-Programmierung erforderlich.					
<b>Lernziele (learning outcomes)</b> Die Teilnehmer können wichtige Klassen von Optimierungsalgorithmen erklären, wie konjugierte Gradienten (CG), BFGS, CMA-ES und den Simplex-Algorithmus für lineare Programmierung. Sie verstehen die Eigenschaften verschiedener Arten von Suchrichtungen (Sampling-Methoden, Gradient, konjugierter Gradient, Newton-Schritt und Quasi-Newton-Methoden) sowie die Mechanismen zur Kontrolle der Schrittweite (Liniensuche und Vertrauensbereich-Methoden). Sie können diese Methoden mit Konvergenzgeschwindigkeitsklassen wie linearer und superlinearer Konvergenz in Verbindung bringen. Sie verstehen Lagranges und können duale Optimierungsprobleme ableiten. Sie können Probleme der realen Welt als mathematische Optimierungsprobleme modellieren. Sie können geeignete Algorithmen auswählen, sie auf reale Optimierungsprobleme anwenden und diese effizient lösen.					
<b>Inhalt</b> Dieser Kurs bietet eine zeitgemäße Einführung in die numerische Optimierung. Optimierungsalgorithmen finden in vielen Bereichen des Ingenieurwesens, der Wirtschaft, des maschinellen Lernens und vielen mehr Anwendung. In diesem Kurs werden die wichtigsten Entwurfsprinzipien und Algorithmenklassen behandelt:  - Gradienten- und Newton-Suchrichtungen  - Liniensuche und Vertrauensbereichsverfahren  - konjugierte Gradienten  - Quasi-Newton-Algorithmen  - Behandlung von Nebenbedingungen, Dualität					

- Lineare Programmierung, einschließlich des gemischt-ganzzahligen Falls

- direkte Suche (gradientenfreie) Methoden

Die Methoden werden in der Vorlesung vorgestellt und analysiert und in den Übungseinheiten implementiert und getestet.

Während der praktischen Sitzungen arbeiten die Teilnehmer an einer Mischung aus konzeptionellen und praktischen Übungen. Viele praktische Übungen beinhalten die Programmierung in Python.

**Lehrformen**

The contact time is split roughly 50/50 into lecture and exercise session. The lecture includes flipped classroom elements. For that purpose, homework assignments include (in some cases extensive) reading assignments.

**Prüfungsformen**

Elektronische Klausur (90 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene schriftliche Modulabschlussprüfung

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/105: M.Sc. Angewandte Informatik

6/97: M.Sc. Computer Science

<b>Titel des Moduls: Privacy Engineering, Data Governance and Usability</b> Privacy Engineering, Data Governance and Usability					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Privacy Engineering, data governace and usability (212037)			<b>Kontaktzeit</b> 45 h	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Dr. Veelasha Moonsamy Lehrende: Dr. Veelasha Moonsamy, Dr. Asia Biega Dr. Yixin Zou					
<b>Verwendung des Moduls</b> M.Sc. IT-Sicherheit/Informationstechnik  M.Sc. IT-Sicherheit/Netze und Systeme					
<b>Vorkenntnisse</b> Recommended but not mandatory: Einführung in die Usable Security and Privacy (211036) Datenschutz (260081); Basic knowledge of threat modeling; General understanding of machine learning and data science					
<b>Lernziele (learning outcomes)</b> By the end of the course, the student will be able to: Reason about privacy concerns and perform threat modelling Apply privacy-by-design techniques for systems implementation Develop privacy technologies Understand concepts related to data governance, including data minimization Design privacy-friendly, usable systems Understand concept related to UX design & usable privacy					
<b>Inhalt</b> This course will provide students with the knowledge and applied skills to tackle the design and implementation of privacy-preserving systems. Students will gain a critical understanding of privacy's role in society and tensions between privacy, technology and security. Students will learn to analyze privacy issues and develop privacy-friendly solutions by considering social, technical, legal and public policy aspects. The course includes mandatory lecture attendance, readings and group project.  The course will cover the following topics: Privacy definitions and concepts Privacy by design Privacy engineering: design and evaluation Data governance Notion of "Right to be forgotten" Usable privacy, including UX design Inclusive privacy					
<b>Lehrformen</b> The course includes mandatory lecture attendance, readings and group project.					
<b>Prüfungsformen</b> There will be one semester-long individual project and a written exam (60 minutes).					

**Voraussetzungen für die Vergabe von Credits**

Final grade: 50% project + 50% exam. You need to pass the exam (i.e. achieve more than 50 points) in order to pass the course.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

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

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

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

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

**Titel des Moduls: Privacy-Preserving Natural Language Processing**  
**Privacy-Preserving Natural Language Processing**

<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Privacy-Preserving Natural Language Processing (211068)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Ivan Habernal  
 Lehrende: Prof. Dr. Ivan Habernal

**Verwendung des Moduls**

**Vorkenntnisse**

Sehr zu empfehlen sind Kenntnisse in moderner natürlicher Sprachverarbeitung (NLP), einschließlich Transformatormodellen, Modelltraining mit SGD usw., siehe z. B. den Kurs Natural Language Processing with Deep Learning, der von Ivan Habernal im WiSe gehalten wird. Sehr solide Kenntnisse der Wahrscheinlichkeitstheorie sind ebenfalls erforderlich. Wir benötigen keine Messwerttheorie, werden aber mit Konzentrationsungleichungen (Markov-Ungleichungen, Chernoff-Schranken usw.) und ähnlichen formalen Methoden arbeiten, die in der differenziellen Privatsphäre verwendet werden.

**Lernziele (learning outcomes)**

Die Studierenden sind in der Lage

- die Schwierigkeiten des Schutzes der Privatsphäre bei der Verwendung nicht-formaler Ad-hoc-Methoden zu verstehen
- die formale Behandlung der Privatsphäre durch algorithmisch randomisierte Ansätze wie die differentielle Privatsphäre zu schätzen wissen
- mit grundlegenden Datenschutzmechanismen formal zu arbeiten
- DP-geschützte Modelle mit Standard-Frameworks zu trainieren

**Inhalt**

Wir werden die folgenden Themen behandeln:

- Warum ist Datenschutz im modernen NLP wichtig?
- Formalisierung des Datenschutzes
- Textanonymisierung, Schwärzung und PII-Entfernung
- Probabilistische Behandlung der Privatsphäre mit differentieller Privatsphäre
- Grundlegende Datenschutzmechanismen (Laplace, Gaussian)
- Training von Modellen mit DP-SGD
- Angriffe auf Modelle (membership inference attacks und ähnliche)
- Kritische Bewertung von DP-Methoden in NLP

**Lehrformen**

Ich habe vor, die Vorlesungen so interaktiv wie möglich zu gestalten, damit alle an Bord bleiben. Rechnen Sie also damit, dass Ihnen Fragen gestellt werden, und Sie sollten sich auch an Dinge aus früheren Vorlesungen erinnern (derselbe Ansatz, den wir in NLP mit DL im WiSe verfolgt haben). Wir werden in den Vorlesungen theoretische Inhalte und einige wichtige formale Beweise behandeln und einige Beweise in die Übungen mitnehmen, um sie zu bearbeiten. In den Übungen werden wir eine Mischung aus Programmieren und der Diskussion von Details zum Datenschutz haben. Bringen Sie also bitte Ihren Laptop mit, auf dem Ihre Lieblings-IDE für Python installiert ist und der bereit ist zu arbeiten. Wir werden auch ein oder zwei Hausaufgaben machen

(die Sie alleine bearbeiten müssen, nicht in Gruppen), die sich mit bestimmten Aspekten des Datenschutzes im NLP beschäftigen; wir werden das während der Übungen im Detail besprechen.

**Prüfungsformen**

Klausur (90 Minuten)

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Product Lifecycle Management**  
**Product Lifecycle Management**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Product Lifecycle Management (138577)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr.-Ing. Detlef Gerhard  
 Lehrende: Prof. Dr.-Ing. Detlef Gerhard

**Verwendung des Moduls**

M.Sc. Angewandte Informatik  
 M.Sc. Maschinenbau  
 M.Sc. Sales Engineering and Product Management

**Vorkenntnisse**

**Lernziele (learning outcomes)**

Nach dem erfolgreichen Abschluss des Moduls

- verfügen Studierende über ein breites, integriertes Wissen über die Herausforderungen des ganzheitlichen Informationsmanagements im Produktlebenszyklus und die resultierenden Anforderungen an Softwaresysteme zur Unterstützung von PLM.
- kennen und verstehen Studierende die Teilprozesse des Produktlebenszyklus, die Methoden des Product Lifecycle Management (PLM) sowie die wissenschaftlichen Grundlagen der zugrundeliegenden IT-Systeme. Indem sie praktische Beispiele und Aufgaben mit entsprechender Anwendungssoftware bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit PLM Software auf konkrete und praxisorientierte PLM Problemstellungen übertragen.
- haben Studierende ein umfassendes Verständnis vom Zusammenwirken der PLM Softwaresysteme und können kritisch die Eignung von Methoden zum Objektmanagement, zum Produktstruktur- und Konfigurationsmanagement sowie zum Projekt- und Prozessmanagement differenzieren und beurteilen.
- können Studierende prozessorientiert an PLM Aufgabenstellungen herangehen, diese reflektieren und bewerten sowie selbstgesteuert verfolgen.
- können Studierende kooperativ PLM-Aufgabenstellungen in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen sowie über Sachverhalte umfassend kommunizieren.

**Inhalt**

Das Modul vermittelt Methoden und Werkzeuge zum Product Lifecycle Management (PLM), insbesondere das dazu erforderliche Grundlagenwissen und die relevanten methodischen Aspekte von Produktinnovationsprozessen. Schwerpunkte bilden dabei die verschiedenen PLM-Funktionen entsprechender Softwaresysteme (z.B. Teile-, Dokumenten- und Produktstrukturmanagement, Klassifizierung, Konfigurationsmanagement, Projekt- und Prozessmanagement). Weiterhin werden allgemeine Methoden zur Organisation und Handhabung von Produktdaten und Benutzerinformationen sowie Methoden des Collaborative Engineerings und die Vorgehensweise bei der PLM-Einführung vermittelt.

**Lehrformen**

Vorlesung mit Übung

**Prüfungsformen**

Klausur 'Modulabschlussprüfung: Product Lifecycle Management' (90 Min., Anteil der Modulnote 100 %)

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

**Voraussetzungen für die Vergabe von Credits**

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

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Quantum Information and Computation [M.Sc.]**  
**Quantum Information and Computation [M.Sc.]**

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

**Modulbeauftragte/r und hauptamtlich Lehrende**

Modulbeauftragte/r: Prof. Dr. Michael Walter  
 Lehrende: Prof. Dr. Michael Walter

**Verwendung des Moduls**

M.Sc. Angewandte Informatik  
 M.Sc. IT-Sicherheit/ Informationstechnik  
 M.Sc. IT-Sicherheit/ Netze und Systeme  
 M.Sc. Computer Science

**Vorkenntnisse**

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

**Lernziele (learning outcomes)**

You will learn fundamental concepts, algorithms, and results in quantum information and computation. After successful completion of this course, you will know the theoretical model of quantum information and computation, how to generalize computer science concepts to the quantum setting, how to design and analyze quantum algorithms and protocols for a variety of computational problems, and how to prove complexity theoretic lower bounds. You will be prepared for an advanced course or a research or thesis project in this area. Master's students will be expected to understand the material in a deeper way, which will reflect itself in homework and examination.

**Inhalt**

This course will give an introduction to quantum information and quantum computation from the perspective of theoretical computer science.

Topics to be covered will likely include:

- Fundamentals of quantum computing: quantum bits, states and operations
- The power of quantum entanglement: nonlocal games
- Entanglement as a resource: superdense coding and teleportation
- Quantum circuit model of computation
- Quantum computing with oracles: Deutsch-Jozsa, Bernstein-Vazirani, Simon
- Quantum Fourier transform and phase estimation
- Shor's factoring algorithm
- Grover's search algorithm and beyond: how to solve SAT on a quantum computer?
- From no cloning to quantum money: a peek at quantum cryptography

The course should be of interest to students of computer science, mathematics, physics, and related disciplines. Students interested in a BSc or MSc project in quantum information, computing, cryptography, etc. are particularly encouraged to participate.

**Lehrformen**

Lecture with Exercise

**Prüfungsformen**

Final written module exam (180 minutes)

**Voraussetzungen für die Vergabe von Credits**

Passed written exam

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

5 /91: M.Sc. IT-Sicherheit/ Informationstechnik

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

5/97: M.Sc. Computer Science

<b>Titel des Moduls: Responsible AI</b> Responsible AI					
<b>Modul-Nr./Code</b>	<b>Credits</b> 6 CP	<b>Workload</b> 180h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Responsible AI (211064)			<b>Kontaktzeit</b> 60h (4 SWS)	<b>Selbststudium</b> 120h	<b>Gruppengröße</b> 50 Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Bilal Zafar Lehrende: Prof. Dr. Bilal Zafar					
<b>Verwendung des Moduls</b>					
<b>Vorkenntnisse</b> Die Teilnehmer sollten einen der folgenden Kurse an der RUB (oder einen ähnlichen Kurs) belegt haben: Machine Learning, (ii) Deep Learning, (iii) Machine Learning: Supervised Methods, (iv) Introduction to Data Science. Die Teilnehmer sollten sehr gut mit der Programmiersprache Python umgehen können.					
<b>Lernziele (learning outcomes)</b> <ul style="list-style-type: none"> <li>• Entwicklung eines Verständnisses dafür, warum Probleme wie Verzerrungen, Toxizität und mangelnde Robustheit in ML-Modellen auftreten, und Erlernen von Techniken zur Bewältigung dieser Probleme.</li> <li>• Lernen Sie die Grenzen von technischen Eingriffen zur Kontrolle von Verzerrungen kennen.</li> <li>• Lernen Sie, wie Sie die Entscheidungen komplexer ML-Modelle und die Fallstricke der Erklärbarkeit erklären können.</li> </ul>					
<b>Inhalt</b> ML-Modelle werden zunehmend in Bereichen eingesetzt, in denen ihre Entscheidungen direkte Auswirkungen auf Menschen haben. So werden ML-Modelle beispielsweise zur Diagnose medizinischer Erkrankungen, zur Abstimmung von Nutzern mit Online-Werbung, zur Bewertung der Kreditwürdigkeit und mancherorts sogar zur Entscheidung über eine Kautionsvorlage vor der Gerichtsverhandlung eingesetzt. Da diese Entscheidungen echte Menschen betreffen, ist es von größter Bedeutung, dass die Entscheidungen der Modelle vertrauenswürdig sind. Zahlreiche Untersuchungen haben jedoch gezeigt, dass diese Modelle mit verschiedenen Problemen behaftet sind. So können die Modelle beispielsweise Menschen aus bestimmten demografischen Gruppen in unfairer Weise benachteiligen, toxische Ergebnisse produzieren, ihre Ergebnisse aufgrund unwichtiger Änderungen der Eingaben erheblich verändern und die Privatsphäre ihrer Nutzer verletzen. In Anbetracht der Komplexität dieser Modelle kann es recht schwierig sein, die Ursachen für diese Probleme zu finden und sie zu beheben.  In diesem Kurs werden Sie lernen, woher diese Vertrauenswürdigkeitsprobleme kommen, wie man sie erkennt und welche Maßnahmen man ergreifen kann, um sie zu beseitigen. Sie werden auch lernen, dass diese Probleme nicht allein durch technische Maßnahmen gelöst werden können, sondern dass oft ein interdisziplinärer Ansatz erforderlich ist, um sie anzugehen.					
<b>Lehrformen</b> Vorlesungen mit anschließenden Aufgaben. Vor jeder Vorlesung wird Ihnen eine Pflichtlektüre zugewiesen, bei der es sich häufig um eine oder mehrere Forschungsarbeiten handelt.					
<b>Prüfungsformen</b> Bewertete Projektarbeit (50% der Note), E-Abschlussklausur (50% der Note) über 120 Minuten					

**Voraussetzungen für die Vergabe von Credits**

Bestehende Note in der Projektarbeit und der e-Klausur.

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/97: M.Sc. Computer Science

6/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Scheduling Theory</b> Scheduling Theory					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Scheduling Theory (212032)			<b>Kontaktzeit</b> 3 SWS (45 h)	<b>Selbststudium</b> 105 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> Keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Steffen Bondorf Lehrende: Prof. Dr.-Ing. Steffen Bondorf					
<b>Verwendung des Moduls</b> M.Sc. Computer Science					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Die Studierenden sollen eine tiefe Kenntnis über Modellierung und Analyse von (harten) Echtzeitsystemen erhalten, insbesondere deterministische Warteschlangentheorie (Networking) sowie Scheduling (Networks, Systems).					
<b>Inhalt</b> Das Scheduling von geteilten Ressourcen (z.B. Forwarding in Netzwerken, CPU-Zeit von lokalen Systemen) spielt eine zentrale Rolle für die Leistungsfähigkeit moderner Systeme. Anforderungen an die Dauer von Datenübertragungen bzw. Berechnungen werden in Echtzeitsystemen in Form von (harten) Deadlines gestellt. Das Einhalten dieser Deadlines muss nachgewiesen bzw. (konstruktiv) garantiert werden. Hierzu stellt die Vorlesung Modellierungs- und Analysemethoden für (verteilte) Echtzeitsysteme vor.					
<b>Lehrformen</b> Die Vorlesung wird als seminaristischer Unterricht abgehalten, die praktischen Übungen am Rechner können weitere Lehrformen wie Gruppen- und Projektarbeit beinhalten.					
<b>Prüfungsformen</b> Klausur (150 Minuten) oder mündliche (15-45 Minuten) Modulabschlussprüfung.					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Modulabschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 5/97: M.Sc. Computer Science					

<b>Titel des Moduls: Simulationstechnik</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Simulationstechnik (127022)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Markus K Lehrende: Prof. Dr. Markus K					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Objektorientierte Modellierung					
<b>Lernziele (learning outcomes)</b> Dieses Modul vermittelt Kompetenzen zum Einsatz von rechnergestützten Simulationskonzepten zur Lösung praxisbezogener Aufgabenstellungen im Bau- und Umweltingenieurwesen. Im Rahmen der Übungsveranstaltungen erfolgt eine Einführung in aktuelle Simulations- und Optimierungssoftware. Die Projektarbeit wird als Gruppenarbeit durchgeführt.					
<b>Inhalt</b> Es werden Vorgehensweisen zur Simulation komplexer Systeme vermittelt. Folgende Themen werden behandelt: <ul style="list-style-type: none"> <li>• Systemanalyse und Modellbildung</li> <li>• System Dynamics</li> <li>• Ereignisdiskrete Simulation</li> <li>• Multiagentensimulation</li> <li>• Aufbereitung von Eingangsdaten</li> <li>• Stochastische Simulation</li> <li>• Simulationsgestützte Optimierung</li> <li>• Einführung in die Software AnyLogic</li> </ul> Innerhalb der Projektarbeit werden aktuelle Fragestellungen aus den Bau- und Umweltingenieurwissenschaften (Montage- und Logistikprozesse, Fußgängersimulation, Schadstoffausbreitung, etc.) aufgearbeitet und mit Hilfe einer objektorientierten Simulationssoftware analysiert. Den Studierenden werden Softwarelizenzen durch den Lehrstuhl zur Verfügung gestellt. Die Programmierung erfolgt unter Verwendung der Programmiersprache Java.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> Projektarbeit mit Abschlusspräsentation					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene Projektarbeit mit Abschlusspräsentation					

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Software Languages**  
**Software Languages**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Software Languages (212034)			<b>Kontaktzeit</b> 4 SWS (60 h)	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> 25 Studierende

<b>Unterrichtssprache</b> Englisch	<b>Teilnahmevoraussetzungen</b>
---------------------------------------	---------------------------------

**Modulbeauftragte/r und hauptamtlich Lehrende**  
 Modulbeauftragte/r: Prof. Dr. Thorsten Berger  
 Lehrende: Prof. Dr. Thorsten Berger

**Verwendung des Moduls**  
 M.Sc. Computer Science  
  
 M.Sc. Angewandte Informatik

**Vorkenntnisse**  
 Objektorientierte Programmierung  
 Theoretische Informatik (insbesondere formale Sprachen, Grammatiken, Mengenschreibweise)  
 Softwaretechnik  
 Funktionale Programmierung

**Lernziele (learning outcomes)**  
**Wissen und Verständnis**

- Erläuterung zentraler Konzepte für die Entwicklung von Softwaresprachen, z. B. Metamodellierung, abstrakte Syntax, konkrete Syntax, statische Semantik, dynamische Semantik, Modell/Programmtransformation
- Erläuterung linguistischer Architekturen für Software-Sprachen (z. B. Metamodellierungshierarchie)
- erklären, wie domänenspezifische Sprachen in einer modernen Sprachwerkbank realisiert werden können

**Kompetenzen und Fertigkeiten**

- entwickeln domänenspezifische Sprachen für ein spezifisches technisches Problem
- Definition einer abstrakten Syntax durch Modellierung von Domänen und Erstellung von Metamodellen
- statische Semantik mit syntaktischen Beschränkungen oder Typsystemen definieren
- Definition einer konkreten Syntax mit Xtext, Sirius oder einer vergleichbaren Technologie
- Definition der dynamischen Semantik durch Modelltransformationen (Kompilierung) oder Interpretation
- Schlussfolgerungen über Konfigurationsräume, die in Variabilitätsmodellen ausgedrückt werden

**Beurteilung und Vorgehen**

- Identifizierung von Anwendungsfällen und des Potenzials der Verwendung von DSLs für eine bestimmte Domäne/Problemstellung
- Auswahl und Begründung einer geeigneten Sprachtechnologie für eine gegebene Domäne/Problemstellung

**Inhalt**  
 Der Kurs lehrt die Theorie und die Pragmatik der Verwendung und Entwicklung von Software-Hochsprachen (Domain-Specific Languages oder DSLs) für die effektive Produktion von Qualitätssoftware. Dazu gehören Methoden, Entwurfsmuster, Richtlinien und Testverfahren zur Definition der konkreten Syntax, der abstrakten

Syntax und der Semantik von Sprachen. Der Kurs versucht, nah an der Technologie zu sein, während er mehrere Paradigmen und Lösungen abdeckt.

Wir behandeln:

- Domänenanalyse einer Problemdomäne und Entwurf von Metamodellen
- Entwicklung externer und interner DSLs (wir konzentrieren uns hauptsächlich auf externe DSLs)
- Entwurf der konkreten Syntax von Sprachen
- Implementierung der Sprachsemantik mit Hilfe von deklarativen und imperativen Transformationen, Codegeneratoren und Interpretern, in verschiedenen Szenarien, z.B. von Text zu Modellen, von Modellen zu Text, unter Einbeziehung von XML, Datenbanken, etc.
- Implementierung von deklarativen Beschränkungen und Typregeln für DSLs
- Testen und Qualitätssicherung von Sprachimplementierungen
- Modellgesteuerte Entwicklung, insbesondere Entwicklung hochgradig konfigurierbarer Systeme oder Software-Produktlinien (wir konzentrieren uns auf Syntax und Semantik der Funktionsmodellierung)

#### **Lehrformen**

Der Unterricht in diesem Kurs besteht aus verschiedenen Formen: Vorlesungen, interaktive Quiz, Gruppenarbeit, Gruppenbetreuung und praktische Aufgaben.

#### **Prüfungsformen**

Die Prüfungsleistung für das Modul Software Languages setzt sich aus mehreren Teilen zusammen. Während des Semesters sind Abgaben zu erbringen, die bestanden werden müssen. Die Abgaben selbst sind unbenotet. Am Ende des Semesters findet eine Klausur oder mündliche Abschlussprüfung statt. Die Voraussetzung, um an der Abschlussprüfung teilnehmen zu können, ist das Bestehen der semesterbegleitenden Abgaben.

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

Bestandene semesterbegleitende Abgaben und bestandene Modulabschlussprüfung (schriftlich (120 Minuten) oder mündlich (15-45 Minuten)).

#### **Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/97: M.Sc. Computer Science

5/105: M.Sc. Angewandte Informatik

**Titel des Moduls: Software Security 1 [M.Sc.]**  
**Software Security 1 [M.Sc.]**

<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Wintersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Software Security 1 (212026)			<b>Kontaktzeit</b> 4 SWS (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. Kevin Borgolte Lehrende: Prof. Kevin Borgolte					
<b>Verwendung des Moduls</b> M.Sc. IT-Sicherheit/ Informationstechnik  M.Sc. IT-Sicherheit/ Netze und Systeme  M.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b> Prior knowledge about programming in Python, C, and assembler is recommended. The following courses (or equivalent) are required: System Security (211011) Operating Systems (211005)					
<b>Lernziele (learning outcomes)</b> At the end of this course, students will be able to: <ul style="list-style-type: none"> <li>• analyze user-space software vulnerability types and protection mechanisms</li> <li>• understand how to write code to reduce the risk of vulnerabilities and apply defensive programming techniques</li> <li>• identify new software vulnerabilities and evaluate their impact</li> <li>• show the existence of a vulnerability, for example, by developing proof of concept verifications</li> </ul>					
<b>Inhalt</b> The course covers the area of introductory software security, vulnerability discovery, and vulnerability verification, focusing on: <ul style="list-style-type: none"> <li>• Assembly and Disassembly, Shellcode</li> <li>• Binary Reverse Engineering and Debugging</li> <li>• Memory and Type Safety/Errors</li> <li>• Stack-based Buffer Overflows</li> <li>• Heap Attacks</li> <li>• Information Leakage</li> <li>• Format String Vulnerabilities</li> <li>• Code Re-use Attacks</li> <li>• Types and Type Safety</li> <li>• Race Conditions</li> </ul>					
<b>Lehrformen</b> Vorlesung mit Übung					

**Prüfungsformen**

Combinated exam: written exam (120 minutes) 40% + practical exercises 60% (both parts need to be passed)

**Voraussetzungen für die Vergabe von Credits**

Passed combinated exam

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

5/91: M.Sc. IT-Sicherheit/ Informationstechnik

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

5/105: M.Sc. Angewandte Informatik

5/91: M.Sc. Computer Science

<b>Titel des Moduls: Studienprojekt (Master)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 10 CP	<b>Workload</b> 300 h	<b>Semester</b> 3	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Individuelle Team- bzw. Gruppenprojekte			<b>Kontaktzeit</b> 30 h	<b>Selbststudium</b> 270 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: Dozierende im Studiengang Angewandte Informatik					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Gute Programmierkenntnisse (aus den Modulen Informatik 1 und 2, Software-Engineering) und ggf. weitere Kompetenzen je nach Wahl des Projektes					
<b>Lernziele (learning outcomes)</b> Ziele des Studienprojekts sind: <ul style="list-style-type: none"> <li>• Die Anwendung des erlernten Fachwissens.</li> <li>• Der Erwerb zusätzlicher Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung.</li> <li>• Die Schulung hinsichtlich der Erarbeitung eigener Lösungsstrategien.</li> <li>• Die Schulung hinsichtlich Arbeitsteilung und Zusammenarbeit im Team (Teamfähigkeit und Projektorganisation).</li> <li>• Erwerb weiterer Kompetenz im Hinblick auf die Dokumentation und die Präsentation von Ergebnissen.</li> </ul>					
<b>Inhalt</b> Im Rahmen des Studienprojekts soll eine Aufgabe aus Bereichen der Angewandten Informatik in Teamarbeit unter Anleitung eines Betreuers gelöst werden. Die angebotenen Projekte decken dabei thematisch die gesamte Bandbreite der Vertiefungsmodule ab. Darüber hinaus werden auch interdisziplinäre Studienprojekte angeboten. In diesen arbeiten z.B. Angewandte Informatiker mit Studierenden der Sozialwissenschaften oder der Sportwissenschaft zusammen.					
<b>Lehrformen</b> Betreutes Projekt					
<b>Prüfungsformen</b> Projektarbeit mit Abschlussbericht					
<b>Voraussetzungen für die Vergabe von Credits</b> Aktive Teilnahme am Projekt, fristgerechte Abgabe des Abschlussberichts und Bewertung des Abschlussberichts mit min. ausreichend.  Hinweis: Jeder Abschlussbericht muss im Umfang von einer DinA4 Seite eine Reflexion über das methodische Vorgehen enthalten.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>  10/105: M.Sc. Angewandte Informatik					

<b>Titel des Moduls: Systemsicherheit</b> System Security					
<b>Modul-Nr./Code</b>	<b>Credits</b> 5 CP	<b>Workload</b> 150 h	<b>Semester</b> siehe Prüfungsordnung	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Systemsicherheit (211011)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 90 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Englisch			<b>Teilnahmevoraussetzungen</b> keine		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Ghassan Karame Lehrende: Prof. Dr. Ghassan Karame					
<b>Verwendung des Moduls</b> B.Sc. IT-Sicherheit/ Informationstechnik  B.Sc. Informatik  M.Sc. Angewandte Informatik  M.Sc. IT-Sicherheit/ Netze und Systeme					
<b>Vorkenntnisse</b> Background in Cryptographic primitives (encryption methods, signatures, MACs, hash functions), principles of communication networks, is recommended.					
<b>Lernziele (learning outcomes)</b> At the end of this course, students will be able to <ul style="list-style-type: none"> <li>• classify and describe vulnerabilities and protection mechanisms of popular systems and protocols, and</li> <li>• analyze / reason about basic protection mechanisms for modern OSs, software, and hardware systems. Students will also develop the ability to reason about the security of a given protocol and independently develop appropriate security defenses and security models.</li> </ul>					
<b>Inhalt</b> While clearly beneficial, the large-scale deployment of online services has resulted in the increase of security threats against existing services. As the size of the global network grows, the incentives of attackers to abuse the operation of online applications also increase and their advantage in mounting successful attacks becomes considerable.  These cyber-attacks often target the resources, availability, and operation of online services. With an increasing number of services relying on online resources, integrating proper security measures therefore becomes integral to ensure the correct functioning of every online service.  In this course, we discuss important theoretical and analytical aspects in system security. The focus of the course is to understand basic attack strategies on modern systems and platforms, with a focus on side-channel attacks, software-based attacks, malware analysis, as well as software-based defenses (e.g., address space randomization and non-executable memory) and hardware-based defenses (e.g., using TPMs and TEEs). Other topics of the course include analyzing the security of modern cryptocurrencies and ML platforms, and similar aspects in system security.  An integral part of this course are exercises and homeworks, which aim to deepen the understanding of the material with practical examples.					
<b>Lehrformen</b>					

**Prüfungsformen****Voraussetzungen für die Vergabe von Credits****Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

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

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

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

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

5/105: M.Sc. Angewandte Informatik

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

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

**Titel des Moduls: Theorie des maschinellen Lernens**  
**Theory of machine learning**

<b>Modul-Nr./Code</b>	<b>Credits</b> 9 CP	<b>Workload</b> 270 h	<b>Semester</b> siehe Prüfungsordnung / see examination regulations	<b>Turnus</b> Sommersemester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b> Theorie des maschinellen Lernens (211052)			<b>Kontaktzeit</b> 6 SWS (90 h)	<b>Selbststudium</b> 180 h	<b>Gruppengröße</b> 20 Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr. Asja Fischer Lehrende: Prof. Dr. Asja Fischer					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc. Computer Science					
<b>Vorkenntnisse</b>					
<b>Lernziele (learning outcomes)</b> Die Studierenden werden mit mathematischen Modellen für das maschinelle Lernen vertraut gemacht. Sie erwerben die Fähigkeit, Lernalgorithmen zu beurteilen und zu vergleichen anhand des Grades, in welchem diese (exakt beschriebene) Erfolgskriterien erreichen. Sie erwerben Techniken sowohl zum Design effizienter Lernalgorithmen als zum Nachweis der inhärenten Härte eines Problems. Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• kennen Studierende die wichtigsten Lernmaschinen (wie zum Beispiel Support Vector Machines und verwandte Modelle),</li> <li>• verstehen Studierende den Unterschied zwischen empirischer und realer Fehlerrate und kennen Techniken zum Umgang mit dem Problem des overfitting der Daten (mit einem zu komplexen Modell),</li> <li>• können Studierende zwischen uniformer und nicht uniformer Lernbarkeit einer Hypothesenklasse unterscheiden und kennen die dazu passenden Theorien und Lernregeln.</li> </ul>					
<b>Inhalt</b> Gegenstand der Vorlesung ist die statistik-basierte Theorie des maschinellen Lernens. Insbesondere wird die Methode der strukturierten Risikominimierung vermittelt sowie die ihr zugrunde liegenden statistischen Lehrsätze. Es werden sowohl Techniken zum Entwurf effizienter Lernalgorithmen besprochen als auch informations- oder berechnungstheoretische Barrieren, die bestimmte Lernprobleme als nicht effizient lösbar erscheinen lassen.					
<b>Lehrformen</b> Vorlesung mit Übung					
<b>Prüfungsformen</b> mündliche Modulabschlussprüfung (15-45 Minuten)					
<b>Voraussetzungen für die Vergabe von Credits</b> Bestandene mündliche Modulabschlussprüfung.					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b>  9/105: M.Sc. Angewandte Informatik					



<b>Titel des Moduls: Verkehrstechnik</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> a) Verkehrssteuerung (128026)  b) Modellierung und Simulation des Verkehrsflusses (128025)			<b>Kontaktzeit</b> 60 h	<b>Selbststudium</b> 120 h	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Deutsch			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Prof. Dr.-Ing. Justin Geistefeldt Lehrende: Prof. Dr.-Ing. Justin Geistefeldt, Prof. Dr.-Ing. Ning Wu					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik  M.Sc. Bau- und Umweltingenieurwesen					
<b>Vorkenntnisse</b> Höhere Mathematik					
<b>Lernziele (learning outcomes)</b> a)  Die Hörer verfügen über das aktuelle technische Wissen und besitzen erweiterte Kenntnisse über die Methoden der verkehrstechnischen Analyse und Steuerung von Knotenpunkten. Sie haben die Fähigkeit, die in der Praxis angewandten Planungstechniken für Lichtsignalanlagen zu verstehen und komplexe Anlagen einschließlich einer Koordinierung praxisgerecht zu entwerfen.  b)  Die Hörer verfügen über differenzierte Kenntnisse der Gesetzmäßigkeiten des Verkehrsflusses auf Straßen. Sie sind in der Lage, wissenschaftliche Beschreibungsmöglichkeiten dieser Gesetzmäßigkeiten zu reflektieren und ihre praktische Anwendbarkeit zu erkennen. Sie haben die Fähigkeit, selbständig Erweiterungen oder Anpassungen von Verkehrsflussmodellen zu entwickeln.					
<b>Inhalt</b> a)  Es werden Methoden der verkehrstechnischen Analyse und Bemessung von Straßenknotenpunkten sowie Steuerungssysteme für Knotenpunkte und die zu ihrem Betrieb erforderlichen Einrichtungen behandelt. Die in der Praxis üblichen Verfahren werden in der Übung an einigen Beispielen veranschaulicht. Dabei werden EDV-Verfahren eingesetzt. Im Einzelnen werden behandelt: Wartezeitermittlung an Knotenpunkten, vorfahrtsgeregelte Knotenpunkte, Festzeitsteuerung von Signalanlagen, Grüne Welle, Koordinierung im Netz, verkehrsabhängige Steuerung einschließlich Signalprogrammablauf, Signaltechnik, Steuerungskriterien.  b)					

Die theoretischen Grundlagen für die Beschreibung des Verkehrsflusses auf Straßen werden mit Hilfe mathematischer Verfahren erarbeitet. Die zu Grunde liegenden Gesetzmäßigkeiten werden hergeleitet. Im Einzelnen werden behandelt: Kenngrößen des Verkehrsablaufs und deren Zusammenhänge, Fundamentaldiagramm, Kapazität, freier Verkehrsfluss, Kontinuumstheorie, Abstandsmodelle, Fahrzeugfolge-theorie, mikroskopische Verkehrsflusssimulation.

**Lehrformen**

Powerpoint-Präsentationen, Tafel, Vorführungen und Übungen am PC

**Prüfungsformen**

Schriftliche Klausur (120 Minuten); über das gesamte Modul.

**Voraussetzungen für die Vergabe von Credits**

Bestandene Klausur

**Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)**

6/105: M.Sc. Angewandte Informatik

<b>Titel des Moduls: Freie Wahlmodule</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> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> abhängig von Veranstaltungswahl	<b>Selbststudium</b> Je nach Veranstaltungswahl	<b>Gruppengröße</b> Studierende
<b>Unterrichtssprache</b> Je nach Veranstaltungswahl			<b>Teilnahmevoraussetzungen</b>		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studienfachberatung Angewandte Informatik Lehrende:					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> abhängig von Veranstaltungswahl					
<b>Lernziele (learning outcomes)</b> Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den freien Wahlfächern					
<b>Inhalt</b> Studierende müssen Veranstaltungen im Gesamtumfang von 15 CP absolvieren.  Je nach Veranstaltungswahl werden unterschiedliche Inhalte vermittelt.  Studierende haben die Möglichkeit unter den Freien Wahlmodulen auch Fächer jenseits der Informatik zu absolvieren und ihre Soft Skills zu erweitern. Z.B. Die freien Wahlfächer erweitern die Soft Skills. Z.B. wird die englische Fachsprache verbessert, in die Grundlagen der Rechtswissenschaften eingeführt oder Grundkenntnisse der Betriebswirtschaft vermittelt. Die Studierenden haben die Möglichkeit eine Auswahl entsprechend der eigenen Interessen zu treffen.					
<b>Lehrformen</b> abhängig von Veranstaltungswahl					
<b>Prüfungsformen</b> abhängig von Veranstaltungswahl					
<b>Voraussetzungen für die Vergabe von Credits</b> abhängig von Veranstaltungswahl					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 0/105 (unbenotet): M.Sc. Angewandte Informatik					

<b>Titel des Moduls: Masterarbeit und Kolloquium (AI)</b>					
<b>Modul-Nr./Code</b>	<b>Credits</b> 30 CP	<b>Workload</b> 900 h	<b>Semester</b> 4	<b>Turnus</b> Jedes Semester	<b>Dauer</b> 1 Semester
<b>Lehrveranstaltungen</b>			<b>Kontaktzeit</b> 15 h	<b>Selbststudium</b> 885 h	<b>Gruppengröße</b> 1 Studierende
<b>Unterrichtssprache</b> Deutsch oder Englisch			<b>Teilnahmevoraussetzungen</b>  <b>M.Sc. Angewandte Informatik PO 22:</b>  Erfolgreich abgeschlossene Module im Umfang von 70 CP.  <b>M.Sc. Angewandte Informatik PO 20:</b>  Erfolgreich abgeschlossene Module im Umfang von 60 CP		
<b>Modulbeauftragte/r und hauptamtlich Lehrende</b> Modulbeauftragte/r: Studiendekan Angewandte Informatik Lehrende: Lehrende im Studiengang Angewandte Informatik					
<b>Verwendung des Moduls</b> M.Sc. Angewandte Informatik					
<b>Vorkenntnisse</b> Abhängig von der Themenwahl.					
<b>Lernziele (learning outcomes)</b> Nach erfolgreichem Abschluss des Moduls: <ul style="list-style-type: none"> <li>• können Studierende selbstständig und fristgerecht ein wissenschaftliches Thema bearbeiten von der Recherche bis zur Dokumentation der Resultate</li> <li>• können Studierende geeignete wissenschaftliche Verfahren und Methoden, die sie im Studium kennengelernt haben, auswählen, anwenden und weiterentwickeln, um ein konkretes Problem zu lösen</li> <li>• können Studierende ihre Ergebnisse kritisch mit dem Stand der Forschung vergleichen und evaluieren</li> <li>• können Studierende ihre eigenen Ergebnisse angemessen in Wort und Schrift darstellen.</li> </ul>					
<b>Inhalt</b>  Es soll eine anspruchsvolle Fragestellung der Angewandten Informatik bearbeitet und dokumentiert werden. Im Anschluss an die Bearbeitung der Masterarbeit werden die Ergebnisse in Form eines Kolloquiumsvortrags mit anschließender Diskussion präsentiert.					
<b>Lehrformen</b> Abschlussarbeit					
<b>Prüfungsformen</b> Masterarbeit und Kolloquiumsvortrag					
<b>Voraussetzungen für die Vergabe von Credits</b> Sowohl die Masterarbeit als auch der Kolloquiumsvortrag müssen bestanden sein.  Der Anteil der Kolloquiumsnote an der Gesamtnote beträgt 10%					
<b>Stellenwert der Note für die Endnote (bei einem Gesamtstudienumfang von 120 ECTS)</b> 30/105: M.Sc. Angewandte Informatik					

