

**Master Studiengang  
Angewandte Informatik**

**PO 13**

**Modulhandbuch**

**Erstellt am: 23. November 2021**

## Wo finde ich Hilfe während des Studiums?

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

Studiengangsw Webseite:  
<https://www.ai.rub.de/>

Studienfachberatung Angewandte Informatik:  
<https://www.ai.rub.de/kontakte/studienberatung.html>

Prüfungsamt Angewandte Informatik:  
<https://www.ai.rub.de/kontakte/pruefungsamt.html>

Fachschaftsrat Angewandte Informatik:  
<https://ai-rub.de/>

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

### Weitere wichtige Kontaktadressen auf dem Campus sind:

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

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

Talentscouts der RUB:  
<https://studium.ruhr-uni-bochum.de/de/die-talentscouts-der-rub>  
Beratung zu Stipendien; hierzu werden extra Workshops angeboten.

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

International Office:  
<http://www.international.rub.de/ausland/index.html.de>  
Beratung zu Studienaufenthalten im Ausland

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

### Studiengangsziele:

Ziel des Masterstudiums ist die Vermittlung von Kenntnissen auf dem Gebiet der Angewandten Informatik, um komplexe Ingenieur Tätigkeiten selbstständig und verantwortlich durchführen zu können. Der Masterstudiengang führt damit zu einer Berufsqualifizierung, die für eine Mitarbeit in Forschung und Entwicklung mit Führungsverantwortung nötig ist. Er vermittelt zudem die notwendigen Kenntnisse für wissenschaftliche Arbeiten auf Promotionsniveau. Durch die Masterprüfung soll festgestellt werden, ob der Kandidat bzw. die Kandidatin fundierte Kenntnisse und die Fähigkeit zur selbstständigen Anwendung anspruchsvoller wissenschaftlicher Methoden erlernt hat. Die Studierenden sollen zur kritischen Einordnung der wissenschaftlichen Erkenntnisse sowie zu verantwortlichem, interdisziplinärem Denken und Handeln befähigt werden und sollen komplexe Probleme der Angewandten Informatik analysieren und Lösungen erarbeiten können. Erweiterte Sprachkenntnisse und Studienaufenthalte im Ausland sind erwünscht.

### Modularisierungskonzept:

Das Studium ist modular aufgebaut. Die Module stellen zeitlich und inhaltlich abgeschlossene Teilqualifikationen dar. Sie haben, abgesehen von wenigen Ausnahmen, einen Workload von mindestens 5 Creditpoints (CP), wobei ein Creditpoint in etwa einer Arbeitsbelastung von 30 Stunden entspricht. In die Berechnung des Workloads fließt neben der Präsenzzeit auch die Zeit für das Selbststudium mit ein (Bearbeitung von Übungsaufgaben, Vor- und Nachbereitung der Vorlesungen, Lesen geeigneter Literatur,?). Im Masterstudium gibt es keine Pflichtmodule. Die im Wahlpflichtbereich angesiedelten Module erweitern das Grundlagenwissen aus dem Bachelorstudium, um den spezialisierteren Vertiefungsmodulen folgen zu können. Insgesamt bietet das Studium zahlreiche Wahlmöglichkeiten für die Studierenden, um ein individuelles Studienprofil zu entwickeln.

### Prüfungsformen

Im Masterstudiengang sind die folgenden Prüfungsformen vertreten: Klausurarbeiten, Prüfungsgespräche, Studienbegleitende Aufgaben, Projektarbeiten, Seminarbeiträge, Praktika, Masterarbeit mit Kolloquiumsbeitrag. Genaueres zu den jeweiligen Prüfungsformen kann in der gültigen Prüfungsordnung nachgelesen werden.

## Studienplan Master Angewandte Informatik PO 13

Nr	Modul	Lehrveranstaltung	Mind. Umfang Modul (LP)	Semester der Modulprüfung	Autom. Anmeldung im 3. FS	Autom. Wiederanmeldung	Bewertung
<b>Wahlpflichtbereich</b>							
1	Wahlpflichtmodule	Fächer aus dem Wahlpflichtkatalog *	20	1-3	Ja	Ja	benotet
<b>Vertiefungsbereich</b>							
2	Vertiefungsmodule	Fächer aus der Vertiefung **	45	1-3	Nein	Ja	benotet
3	Seminar	Masterseminar ***	3	1-3	Nein	Nein	benotet
4	Freie Wahlfächer	frei wählbar ****	8	1-3	Nein	Nein	unbenotet
			≥4	aus den obigen Bereichen			
<b>Studienprojekt</b>							
5	Master-Studienprojekt	Master-Studienprojekt	10	1-3	Nein	Nein	benotet
<b>Masterarbeit</b>							
6	Masterarbeit und Kolloquium	Masterarbeit und Kolloquium	30	4	Nein	Nein	benotet
Summe:			120				

- \* Die Liste der wählbaren Wahlpflichtmodule befindet sich im jeweils aktuellen Modulhandbuch, welches vom Gemeinsam beschließenden Ausschuss verabschiedet wird.  
Es müssen Module im Umfang von mindestens 20 LP gewählt werden.
- \*\* Die Liste der wählbaren Vertiefungsmodule befindet sich im jeweils aktuellen Modulhandbuch, welches vom Gemeinsam beschließenden Ausschuss verabschiedet wird.  
Es müssen Module im Umfang von mindestens 45 LP gewählt werden.
- \*\*\* Die Liste der wählbaren Masterseminare befindet sich im jeweils aktuellen Modulhandbuch, welches vom Gemeinsam beschließenden Ausschuss verabschiedet wird.  
Es müssen Seminare im Umfang von mindestens 3 LP gewählt werden.
- \*\*\*\* Hier können nahezu alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der UAMR gewählt werden.  
Es müssen Veranstaltungen im Umfang von mindestens 8 LP gewählt werden. Aus der Fakultät Wirtschaftswissenschaften sind nicht alle Veranstaltungen für die AI geöffnet. Eine entsprechende Auflistung befindet sich im aktuellen Modulhandbuch.

Lehrveranstaltung	Einheit	Umfang Modul (LP)	Semester	Autom. Anmeldung	Autom. Wiederanmeldung	Bewertung
<b>Wahlpflichtmodule</b>						
Effiziente Algorithmen	Mathe	9	SS	Nein	Ja	benotet
Groupware und Wissensmanagement	AW	5	SS	Nein	Ja	benotet
Komplexitätstheorie (erst wieder SS 22)	Mathe	9	unregelmäßig	Nein	Ja	benotet
Parallel Computing	Baulng	6	SS	Nein	Ja	benotet
Nebenläufige Programmierung	extern	5	SS	Nein	Ja	benotet
Machine Learning: Supervised Methods	INI	6	SS	Nein	Ja	benotet
Machine Learning: Unsupervised Methods	INI	6	WS	Nein	Ja	benotet
<b>Seminare</b>						
Ingenieurinformatik	MB/Baulng	3	WS/SS	Nein	Nein	benotet
Kryptologie und Theoretische Informatik (Angebot siehe Vorlesungsverzeichnis; im WS 21/22: Seminar Satisfiability)	Mathe	3	unregelmäßig	Nein	Nein	benotet
Seminar (Geometrische) Algorithmen	Mathe	3	unregelmäßig	Nein	Nein	benotet
Computerlinguistik (Angebot siehe Vorlesungsverzeichnis)	Philologie	3	WS/SS	Nein	Nein	benotet
Bioinformatik	PURE/ProDi	3	WS/SS	Nein	Nein	benotet
Knowledge Graphs	Mathe	3	WS	Nein	Nein	benotet
Seminar Deep Learning	Mathe	3	WS	Nein	Nein	benotet
Topics in deep learning for sequence processing	INI	3	WS	Nein	Nein	benotet
Student Conference	ETIT	6	WS /SS	Nein	Nein	benotet
<b>Vertiefungsmodule</b>						
<b>Ingenieurinformatik</b>						
Design Optimization	Baulng	6	WS	Nein	Ja	benotet
Design soziotechnischer Informationssysteme	AW	5	WS	Nein	Ja	benotet
IT im Engineering	MB	6	WS	Nein	Ja	benotet
Product Lifecycle Management	MB	6	SS	Nein	Ja	benotet
Grundlagen der Verkehrsplanung und Verkehrstechnik	Bauing	5	WS	Nein	Ja	benotet
Verkehrstechnik	Bauing	6	SS	Nein	Ja	benotet
Wissensbasierte Methoden	Baulng	6	SS	Nein	Ja	benotet
Grundlagen der automatischen Spracherkennung	ETIT	6	SS	Nein	Ja	benotet
<b>Programmier- und Simulationstechnik</b>						
Fundamentals of GPU Programming	ETIT	5	WS	Nein	Ja	benotet
Simulationstechnik	Baulng	5	WS	Nein	Ja	benotet
3D-Simulation in der Automatisierungstechnik	MB	6	WS	Nein	Ja	benotet
Complexity Economics and Agent-based Modeling	WiWi	10	WS	Nein	Ja	benotet
High-Performance Computing on Clusters	Baulng	6	WS	Nein	Ja	benotet
High-Performance Computing on Multi- and Manycore Processors	Baulng	6	SS	Nein	Ja	benotet
Knowledge Graphs	Mathe	5	SS	Nein	Ja	benotet
<b>Neuroinformatik</b>						
Autonomous Robotics: Action, Perception and Cognition	INI	6	SS	Nein	Ja	benotet
Computational Neuroscience: Neural Dynamics	INI	6	WS	Nein	Ja	benotet
Computational Neuroscience: Vision and Memory	INI	5	SS	Nein	Ja	benotet
Machine Learning: Evolutionary Algorithms	INI	6	WS	Nein	Ja	benotet
Master-Praktikum: Autonomous Robotics	INI	3	WS/SS	Nein	Ja	benotet
Deep Learning	Mathe	5	WS	Nein	Ja	benotet
<b>Kryptologie und Theoretische Informatik</b>						
Kryptanalyse 1	Mathe	5	SS	Nein	Ja	benotet
Geometrische Algorithmen	Mathe	9 bzw. 6 im WS 21/22	WS	Nein	Ja	benotet
Kryptographie	Mathe	8	WS	Nein	Ja	benotet
Kryptographische Protokolle	Mathe	5	SS	Nein	Ja	benotet
Quantenalgorithmen (nicht im SoSe 21)	Mathe	6	SS	Nein	Ja	benotet
Symmetrische Kryptanalyse	Mathe	5	WS	Nein	Ja	benotet
Systemsicherheit	ETIT	5	SS	Nein	Ja	benotet
Model Checking (nicht im WS 22/23, ab SS 23 jeweils im SS)	Mathe	5	WS	Nein	Ja	benotet

<b>Wirtschaftsinformatik &amp; Management</b>							
Industrielles Kundenmanagement	MB	5	SS	Nein	Ja	benotet	
Managing Digital Platform Ecosystems	TUDo	7	SS	Nein	Ja	benotet	
<b>Bioinformatik</b>							
Bioinformatik in der Proteomik I	MPC	5	WS	Nein	Ja	benotet	
Bioinformatik in der Proteomik II	MPC	5	SS	Nein	Ja	benotet	
Bioimage Informatics	Biologie	5	WS	Nein	Ja	benotet	
Master-Praktikum: Big Data in der Bioinformatik	Biologie	5	WS/SS	Nein	Ja	benotet	
Master-Praktikum: Computational Proteomics	MPC	5	WS/SS	Nein	Ja	benotet	

## Abkürzungen:

AW: Institut für Arbeitswissenschaften  
 Baulng: Fakultät für Bau- und Umweltingenieurwissenschaften  
 ETIT: Fakultät für Elektrotechnik und Informationstechnik  
 INI: Institut für Neuroinformatik  
 MB: Fakultät für Maschinenbau  
 MPC: Medizinisches Proteom Center  
 WIWI: Fakultät für Wirtschaftswissenschaft  
 TUDo: Technische Universität Dortmund

SS: Sommersemester

WS: Wintersemester

LP: Leistungspunkte (auch Creditpoints oder CP)

# Inhaltsverzeichnis

<b>1</b>	<b>Wahlpflichtmodule</b>	<b>1</b>
1.1	Effiziente Algorithmen . . . . .	1
1.2	Groupware und Wissensmanagement . . . . .	3
1.3	Komplexitätstheorie . . . . .	5
1.4	Parallel Computing . . . . .	7
1.5	Nebenläufige Programmierung . . . . .	9
1.6	Machine Learning: Supervised Methods . . . . .	12
1.7	Machine Learning: Unsupervised Methods . . . . .	14
<b>2</b>	<b>Masterarbeit und Kolloquium</b>	<b>16</b>
2.1	Masterarbeit . . . . .	16
<b>3</b>	<b>Ingenieurinformatik</b>	<b>18</b>
3.1	Design Optimization . . . . .	18
3.2	Design sozio-technischer Informationssysteme . . . . .	20
3.3	IT im Engineering . . . . .	22
3.4	Künstliche Intelligenz für Ingenieure . . . . .	24
3.5	Product Lifecycle Management . . . . .	26
3.6	Grundlagen der Verkehrsplanung und Verkehrstechnik . . . . .	28
3.7	Verkehrstechnik . . . . .	30
3.8	Wissensbasierte Methoden . . . . .	32
3.9	Grundlagen der automatischen Spracherkennung . . . . .	34
<b>4</b>	<b>Programmier- und Simulationstechnik</b>	<b>36</b>
4.1	Fundamentals of GPU Programming . . . . .	36
4.2	Simulationstechnik . . . . .	38
4.3	3D-Simulation in der Automatisierungstechnik . . . . .	40
4.4	Complexity Economics and Agent-based Modeling . . . . .	42
4.5	High Performance Computing on Clusters . . . . .	44
4.6	High-Performance Computing on Multi- and Manycore Processors . . . . .	46
4.7	Knowledge Graphs . . . . .	48
<b>5</b>	<b>Neuroinformatik</b>	<b>50</b>
5.1	Autonomous Robotics: Action, Perception and Cognition . . . . .	50
5.2	Computational Neuroscience: Neural Dynamics . . . . .	52
5.3	Computational Neuroscience: Vision and Memory . . . . .	54
5.4	Machine Learning: Evolutionary Algorithms . . . . .	56
5.5	Master-Praktikum: Autonomous Robotics . . . . .	58

5.6	Deep Learning . . . . .	60
<b>6</b>	<b>Kryptologie und Theoretische Informatik</b>	<b>61</b>
6.1	Kryptanalyse 1 (Einführung in die asymmetrische Kryptanalyse) . . . . .	61
6.2	Symmetrische Kryptoanalyse . . . . .	63
6.3	Geometrische Algorithmen . . . . .	64
6.4	Kryptographie . . . . .	66
6.5	Kryptographische Protokolle . . . . .	68
6.6	Systemsicherheit . . . . .	70
6.7	Model Checking . . . . .	71
<b>7</b>	<b>Wirtschaftsinformatik und Management</b>	<b>73</b>
7.1	Industrielles Kundenmanagement . . . . .	73
7.2	Rationales Entscheiden . . . . .	76
7.3	Managing Digital Platform Ecosystems . . . . .	78
<b>8</b>	<b>Bioinformatik</b>	<b>80</b>
8.1	Bioinformatics for Proteomics I . . . . .	80
8.2	Bioinformatics for Proteomics II . . . . .	83
8.3	Bioimage Informatics . . . . .	86
8.4	Master-Praktikum Big Data in der Bioinformatik . . . . .	88
8.5	Master-Praktikum Computational Proteomics . . . . .	90
<b>9</b>	<b>Masterseminare</b>	<b>92</b>
9.1	Seminar Ingenieurinformatik . . . . .	92
9.2	Seminar Kryptologie und Theoretische Informatik . . . . .	94
9.3	Seminar Computerlinguistik . . . . .	96
9.4	Seminar Operations Research (Management Science) . . . . .	98
9.5	Seminar Bioinformatik . . . . .	100
9.6	Seminar Algorithmen . . . . .	102
9.7	Seminar Model Checking . . . . .	103
9.8	Seminar Topics in deep learning for sequence processing . . . . .	105
9.9	Student Conference: Research Project . . . . .	107
<b>10</b>	<b>Studienprojekt</b>	<b>109</b>
10.1	Master-Studienprojekt . . . . .	109
<b>11</b>	<b>Freie Wahlfächer</b>	<b>111</b>
11.1	Freie Wahlfächer . . . . .	111



# Wahlpflichtmodule

<b>Effiziente Algorithmen</b>					
<b>Modul-Nr:</b> 150320 (Übung: 150321)	<b>Credits:</b> 9 CP	<b>Workload:</b> 270 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung  b) Übung			<b>Kontaktzeit:</b> V: 4 SWS; Ü: 2	<b>Selbststudium:</b> 180 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Die Inhalte der Veranstaltung "Datenstrukturen".					
<b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Moduls: Die Studierende					
<ul style="list-style-type: none"> <li>• Kennen, auswählen und nutzen grundlegende Datenstrukturen und Graphenalgorithmen</li> <li>• Sind in der Lage Analysetechniken (Korrektheitsbeweise und Laufzeitanalyse) zu erläutern und 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>Vermittelte Kompetenzen:</b> Kernkompetenzen:					
<ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> </ul>					

## KAPITEL 1. WAHLPFLICHTMODULE

Fachspezifische Kompetenzen:

- Algorithmik
- Graphentheorie
- Mathematik
- Optimierungsmethoden

### **Inhalt:**

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:

- Berechnung kürzester Pfade in Digraphen
- Berechnung eines maximalen Flusses in einem Transportnetzwerk
- Berechnung einer optimalen Lösung bei einem Zuordnungsproblem (auch Matching-Problem genannt)

Darüberhinaus beschäftigen wir uns mit Anwendungen dieser grundlegenden Probleme.

### **Lehrformen:**

Vortrag der Lehrenden in der Vorlesung, Gruppenarbeit in den Übungen

### **Prüfungsformen:**

Klausur (120 Minuten)

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Stellenwert der Note für die Endnote:** 9 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**PD Dr. Daniela Kacso**

<b>Groupware und Wissensmanagement</b>					
<b>Modul-Nr:</b> 260080	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung und Übung integriert			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> 18
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Mensch-Maschine-Interaktion</p>					
<p><b>Lernziele (learning outcomes):</b> Insbesondere durch die Nutzung neuer Technologien sollen die Studierenden erfahren, wie „Groupware und Wissensmanagement“ in der Praxis funktioniert und Konzepte erarbeiten, wie die Technik, aber auch die dazugehörigen sozio-technischen Prozesse gestaltet werden müssen, um für die Nutzer/innen nutzbar und nützlich zu sein. Dieser Einsatz ändert sich jeweils im Vergleich zum Vorjahr, da er von der Art der Projektarbeit abhängt.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Projekt- und Zeitmanagement</li> <li>• Teamarbeit und Teamfähigkeit</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Entwicklungsprozesse</li> </ul>					
<p><b>Inhalt:</b></p> <p>In zunehmendem Maße kommen Computersysteme nicht als Individualplatzlösung, sondern in Netzwerken zum Einsatz. Diese unterstützen gemeinsames Arbeiten und Lernen und werden unter den Schlagworten Groupware und Wissensmanagement zusammengefasst. Bei der Gestaltung und Einführung solcher Systeme wird man sich immer auch mit Fragen der Organisations- und Kommunikationsveränderungen befassen müssen. Die Vorlesung ist in vier thematische Blöcke unterteilt, die aufeinander aufbauen:</p> <ul style="list-style-type: none"> <li>• Interdisziplinäre Grundlagen zum Verständnis von menschlicher Kommunikation, Kooperation und Organisation</li> <li>• Groupwaresysteme sowie Methoden und Werkzeuge</li> <li>• Mechanismen zur Kommunikationsunterstützung sowie zur Kooperation und Koordination</li> </ul>					

## KAPITEL 1. WAHLPFLICHTMODULE

- Social Software/Web 2.0,
- Computer supported collaborative Learning
- Wissensmanagement in Organisationen

**Lehrformen:**

Abwechslung von Vorlesungsinhalten, Seminarbeiträgen der Studierenden, Präsenzübungen und Erörterung der Projektbearbeitung

**Prüfungsformen:**

Semesterbegleitend; eigener Seminarbeitrag (40 Minuten + 20 Minuten Diskussion)

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandener Seminarbeitrag (100% der Endnote)

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Thomas Herrmann**  
Markus Jelonek Msc.

**Sonstige Informationen:****Literatur:**

1. Schwabe, G., Streit, N., Unland, R. (Hrsg.) (2001): CSCW Kompendium. Lehr- und Handbuch zum computergestützten kooperativen Arbeiten. Heidelberg u.a.: Springer. S. 25 – 35.
2. Haake, Jörg; Schwabe, Gerd; Wessner, Martin (Hrsg.) (2012): CSCL-Kompendium 2.0: Lehr- und Handbuch zum computerunterstützten kooperativen Lernen. München: Oldenbourg Verlag.

<b>Komplexitätstheorie</b>					
<b>Modul-Nr:</b> 150262 (Übung: 150263)	<b>Credits:</b> 9 CP	<b>Workload:</b> 270 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> unregelmäßig	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung Komplexitätstheorie  b) Uebungen zur Vorlesung			<b>Kontaktzeit:</b> 6 SWS	<b>Selbststudium:</b> 180 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Vorlesung über Theoretische Informatik					
<b>Lernziele (learning outcomes):</b> Die Studierenden lernen, algorithmische Probleme bezüglich ihrer Komplexität einzuordnen und so geeignete algorithmische Techniken zu ihrer Lösung zu identifizieren. Sie können insbesondere algorithmische Methoden für NP-vollständige Probleme anwenden. Sie können mit unterschiedlichen Berechnungsmodellen umgehen und sind in der Lage, einfache Aussagen über sie zu beweisen. Sie lernen im Diskurs eigene und fremde Lösungsansätze zu bewerten.					
<b>Inhalt:</b> Die Komplexitätstheorie untersucht und klassifiziert Berechnungsprobleme bezüglich ihrer algorithmischen Schwierigkeit. Ziel ist es, den inhärenten Ressourcenverbrauch bezüglich verschiedener Ressourcen wie Rechenzeit oder Speicherplatz zu bestimmen, und Probleme mit ähnlichem Ressourcenverbrauch in Komplexitätsklassen zusammenzufassen. Die bekanntesten Komplexitätsklassen sind sicherlich P und NP, die die in polynomieller Zeit lösbaren bzw. verifizierbaren Probleme umfassen. Die Frage, ob P und NP verschieden sind, wird als eine der bedeutendsten offenen Fragen der theoretischen Informatik, ja sogar der Mathematik, angesehen. P und NP sind jedoch nur zwei Beispiele von Komplexitätsklassen. Andere Klassen ergeben sich unter anderem bei der Untersuchung der des benötigten Speicherplatzes, der effizienten Parallelisierbarkeit von Problemen, der Lösbarkeit durch zufallsgesteuerte Algorithmen, und der approximativen Lösbarkeit von Problemen. Die Vorlesung hat das Ziel, einen breiten Überblick über die grundlegenden Konzepte und Resultate der Komplexitätstheorie zu geben:  <ul style="list-style-type: none"> <li>• Klassische Resultate für Platz- und Zeitkomplexitätsklassen: z.B. die Korrespondenz zwischen Spielen und Speicherplatz-Beschränkungen, der Nachweis, dass sich mit mehr Platz oder Zeit auch mehr Probleme lösen lassen, weitere grundlegende Beziehungen zwischen Zeit- und Platzbasierten Klassen, und die Komplexitätswelt zwischen NP und PSPACE</li> <li>• Grundzüge der Komplexitätstheorie paralleler, zufallsbasierter und approximativer Algorithmen</li> <li>• Einführung in ausgewählte neuere Themen: Komplexitätstheorie des interaktiven Rech-</li> </ul>					

## KAPITEL 1. WAHLPFLICHTMODULE

nens, des probabilistischen Beweisens und Fine-grained Complexity.
<b>Lehrformen:</b> Vorlesung mit begleitenden Übungen
<b>Prüfungsformen:</b> Abschlussprüfung; mündlich, 20-30min
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> bestandene mündliche Prüfung
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Studiengang Mathematik
<b>Stellenwert der Note für die Endnote:</b> 9 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Jun.-Prof. Thomas Zeume</b>
<b>Sonstige Informationen:</b> Erst wieder SS 22!  <b>Literatur:</b> Einstiegsliteratur für diese Veranstaltung sind die Bücher: <ul style="list-style-type: none"><li>• Wegener. Komplexitätstheorie: Grenzen der Effizienz von Algorithmen. Springer. 2003.</li><li>• Arora, Barak. Computational Complexity: A Modern Approach. Cambridge University Press. Eine Vorabversion ist verfügbar unter: <a href="http://theory.cs.princeton.edu/complexity/book.pdf">http://theory.cs.princeton.edu/complexity/book.pdf</a></li><li>• Papadimitriou. Computational Complexity. Addison-Wesley. Reading. 1995.</li><li>• Kozen. Theory of Computation. Springer. 2006.</li></ul>

<b>Parallel Computing</b>					
<b>Modul-Nr:</b> 127501	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Parallel Computing			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Programmierkenntnisse in den Sprachen C, C++ oder Java					
<p><b>Lernziele (learning outcomes):</b> Goals of the course “Parallel Computing” include a basic understanding of the theoretical foundation for efficient parallel algorithms, the architecture of parallel hardware systems and current parallel software paradigms used in research and industry today. Also, students learn how to develop parallel algorithms and implement them using state-of-the-art software systems and programming tools. In particular, numerically intensive engineering applications are a focus within the lecture itself as well as a target for the team projects carried out by the students of this course. Further, students must clearly present their projects results in a classroom setting to an audience with various technical background (the course is attended by students from applied computer science as well as by students from computational engineering).</p> <p>When the students have successfully complete this course,</p> <ul style="list-style-type: none"> <li>• students will be familiar with the development of parallel algorithms and parallel software systems and have a good understanding of how to realize efficient parallel programs;</li> <li>• students will have a good understanding of the basics of parallel programming to be able to, for example, select a proper parallel software system or tool that is appropriate for a given numerical engineering task, or be able to discuss possible solutions concerning necessary hardware or software requirements at a competent level.</li> </ul>					
<p><b>Inhalt:</b></p> <ul style="list-style-type: none"> <li>• Introduction to parallel computing using examples from the field of engineering</li> <li>• Theoretical foundations of parallel computing (concurrency, parallel processes, deadlocks, Amdahl’s Law, Flynn’s taxonomy, efficiency metrics, memory models, etc.)</li> <li>• Parallel programming based on “shared memory” using the OpenMP application programming interface</li> <li>• Parallel programming based on “distributed memory” based the Message Passing Interface (MPI)</li> </ul>					

## KAPITEL 1. WAHLPFLICHTMODULE

- Hardware based parallel programming based on the General Purpose Graphical Processing Unit (GPGPU)
- Application of parallel programming paradigms to solve engineering tasks as a team project

### **Lehrformen:**

- lectures
- exercises using computers
- team projects

### **Prüfungsformen:**

Semesterbegleitend; - completion of a team project (2-4 students) - presentation of project results - oral examination

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

- successful completion of a team project
- presentation of project results in the classroom
- oral examination

**Stellenwert der Note für die Endnote:** 6 / 112

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

**Prof. M. König, Dr. K. Lehner**

### **Sonstige Informationen:**

#### **Literatur:**

1. A. Schill, Th. Springer, "Verteilte Systeme", Springer-Verlag, 2007
2. Th. Raubner, G. Rünger, "Parallele Programmierung", Springer-Verlag, 2007
3. G. Bengel, et. al., "Masterkurs Parallele und Verteilte Systeme", Vieweg+Teubner, 2008



<b>Nebenläufige Programmierung</b>					
<b>Modul-Nr:</b> 310509	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung: 2 SWS  b) Übung: 1 SWS			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> 30 Studierende
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Informatik 1, Informatik 2, Web-Engineering, Software-Engineering					
<p><b>Lernziele (learning outcomes):</b> Die Veranstaltung vermittelt grundlegende Fähigkeiten und Techniken, um nebenläufige Programme sicher entwickeln zu können. Es werden softwaretechnische Entwurfsmuster behandelt und vertieft, welche bekannte Probleme bei nebenläufigen Programmen wie zum Beispiel die Verklemmung vermeiden lassen. Der Zuhörer sollte am Ende der Veranstaltung nach einem erfolgreichen Abschluss der Klausur unter anderem in der Lage sein:</p> <ul style="list-style-type: none"> <li>• Die Performanz von Programmen durch den Einsatz der nebenläufigen Programmierung zu verbessern</li> <li>• Bestehende Programme zu analysieren und mögliche Fehler zu erkennen</li> <li>• Die Sprachmerkmale und Schnittstellen von JAVA für die nebenläufige Programmierung sicher anzuwenden</li> </ul> <p>Fach- und Methodenkompetenz im Detail. Die Studierenden sind in der Lage,</p> <ul style="list-style-type: none"> <li>• die Begriffe Anwendung, Programm, Modul, Prozess und Thread voneinander zu unterscheiden (Wissen/Verstehen),</li> <li>• zu verstehen, warum Threads in manchen Situationen synchronisiert werden müssen (Wissen/Verstehen),</li> <li>• Synchronisation als Ursache von Verklemmungen zu begreifen (Wissen/Verstehen),</li> <li>• Verklemmungen durch Anwendung geeigneter Entwurfsmuster zu vermeiden (Erschaffen/Kreieren),</li> <li>• wichtige Klassen des Java APIs in eigenen nebenläufigen Programmen nutzen können (Anwenden),</li> <li>• die UML-Notation zur Modellierung nebenläufiger Sachverhalte zu beherrschen (Wissen/Verstehen sowie Erschaffen/Kreieren)</li> </ul>					

### **Vermittelte Kompetenzen:**

Fachspezifische Kompetenzen:

- Algorithmik
- Software-Entwurfsmethoden
- Programmieren

Technologien:

- CUDA
- Java
- UML

### **Inhalt:**

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

- Nebenläufigkeit: Schnelleinstieg
  - Anwendungen vs. Prozesse
  - Programme und ihre Ausführung
  - Vorteile und Probleme von nebenläufigen Programmen
    - \* Verbesserung der Performanz
    - \* Synchronisation
    - \* Realisierung kritischer Abschnitte
    - \* Monitore
    - \* Lebendigkeit
    - \* Verklemmungen
- Threads in Java
- UML-Modellierung von Nebenläufigkeit
- Neues zur Nebenläufigkeit in Java 5 und Java 6
- Realisierung von Nebenläufigkeit
- Fortgeschrittene Java-Konzepte für Nebenläufigkeit
- Synchronisationsobjekte

## KAPITEL 1. WAHLPFLICHTMODULE

<ul style="list-style-type: none"><li>• Datenstrukturen und Algorithmen für nebenläufige Programme</li><li>• Entwurfsmuster</li></ul>
<b>Lehrformen:</b> Vorlesung mit Medieneinsatz (Powerpoint/Projektion) und geringem Tafelanschrieb
<b>Prüfungsformen:</b> Schriftliche Klausurarbeit (90min)
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Masterstudiengang IT-Sicherheit/Informatik
<b>Stellenwert der Note für die Endnote:</b> 5 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Studiendekan AI</b> Dr. Doga Arinir
<b>Sonstige Informationen:</b> Das Modul setzt grundlegende Kompetenzen im Bereich der Programmierung voraus. Kenntnisse in der Java-,C++- oder C#-Programmierung sind empfehlenswert, werden aber nicht vorausgesetzt.

<b>Machine Learning: Supervised Methods</b>					
<b>Modul-Nr:</b> 310508 (Übung: 310518)	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> flipped classroom			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig, zuletzt 80 Teilnehmer
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> 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> flipped classroom					
<b>Prüfungsformen:</b> Klausur 90 Minuten, Wiederholungsklausur nach 6 Monaten					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Master computational engineering Master Medizinphysik Master Physik Master Biologie Sämtliche Ingenieurstudiengänge Die Veranstaltung wird auch regelmäßig von Studierenden der Chemie, der Mathematik, aus cognitive science, von Doktoranden, sowie von Studierenden der TU Dortmund (insbesondere automation and robotics) nachgefragt.					

<b>Stellenwert der Note für die Endnote: 6 / 112</b>
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Jun.-Prof. Dr. Tobias Glasmachers</b>
<b>Sonstige Informationen:</b> Es bietet sich an, diese Vorlesung in Kombination mit der Veranstaltung "Machine Learning: Unsupervised Methods", angeboten jeweils im Wintersemester, zu hören.

<b>Machine Learning: Unsupervised Methods</b>					
<b>Modul-Nr:</b> 310003 (Übung: 310013)	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Machine Learning: Unsupervised Methods			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> 30
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> The mathematical level of the course is mixed but generally high, including calculus (functions, derivatives, integrals, differential equations, ...), linear algebra (vectors, matrices, inner product, orthogonal vectors, basis systems, ...), and a bit of probability theory (probabilities, probability densities, Bayes' theorem, ...). Programming is done in Python, thus the students should have a basic knowledge of that as well, or at least be fluent in another programming language.</p>					
<p><b>Lernziele (learning outcomes):</b> After the successful completion of this course the students:</p> <ul style="list-style-type: none"> <li>• know a number of important unsupervised learning methods,</li> <li>• can discuss and decide which of the methods are appropriate for a given data set,</li> <li>• understand the mathematics of these methods,</li> <li>• know how to implement and apply these methods in python,</li> <li>• have gained experience in organizing and working in a team,</li> <li>• know problem solving strategies like brain storming,</li> <li>• can communicate about all this in English.</li> </ul>					
<p><b>Inhalt:</b> This course covers a variety of unsupervised methods from machine learning such as principal component analysis, independent component analysis, vector quantization, clustering, Bayesian theory and graphical models.</p>					
<p><b>Lehrformen:</b> This course is given with the problem based learning concept. The students work in groups of about 5 on realistic problems that can be solved with unsupervised learning methods from machine learning. They develop hypotheses and strategies for a solution and identify what they need to learn in order to implement these. Thus the students will not only learn about machine learning but also soft skills.</p>					
<p><b>Prüfungsformen:</b> Oral exam, ca 20 min</p>					

## KAPITEL 1. WAHLPFLICHTMODULE

<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Exam prerequisite and passed oral exam.
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. Elektrotechnik und Informationstechnik
<b>Stellenwert der Note für die Endnote:</b> 6 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Laurenz Wiskott</b>
<b>Sonstige Informationen:</b> There is a lecture, which provides the content, and a tutorial, where you solve exercises and can deepen your understanding of the content. The exercises are solved in the tutorial in a group effort, not at home, which is the reason why it takes 3 hours rather than the usual 1.5 hours.  Course Material: Please download all material you need until end of July, because beginning in August I prepare the new course and some links of the old course might break.

# Masterarbeit und Kolloquium

<b>Masterarbeit</b>					
<b>Modul-Nr:</b> N/A	<b>Credits:</b> 30 CP	<b>Workload:</b> 900 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Jedes Semester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Praktikum/Projektarbeit			<b>Kontaktzeit:</b> 0 SWS	<b>Selbststudium:</b> 900 h	<b>Gruppen- größe:</b> 1
<p><b>Teilnahmevoraussetzungen:</b> 80 LP und abgeschlossener Wahlpflichtbereich, siehe Prüfungsordnung</p> <p><b>Empfohlene Vorkenntnisse:</b> Fachspezifisch von der Themenstellung abhängig.</p>					
<p><b>Lernziele (learning outcomes):</b> Nach erfolgreichem Abschluss des Moduls:</p> <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> <p><b>Vermittelte Kompetenzen:</b> Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Kritikfähigkeit</li> <li>• Projekt- und Zeitmanagement</li> <li>• Präsentation wissenschaftlicher Ergebnisse</li> <li>• Rhetorik und sprachliche Kompetenz</li> </ul>					



## KAPITEL 2. MASTERARBEIT UND KOLLOQUIUM

- Fächerübergreifendes Denken und Arbeiten
- Wissenschaftliches Arbeiten und Schreiben

**Inhalt:**

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.

**Lehrformen:**

Praktikum/Projektarbeit

**Prüfungsformen:**

Abschlussarbeit

**Voraussetzungen für die Vergabe von Kreditpunkten:**

erfolgreiche schriftliche Masterarbeit + Kolloquiums-vortrag

**Stellenwert der Note für die Endnote:** 30 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Studeindegan der Angewandten Informatik**  
Dozenten der RUB

**Sonstige Informationen:**

Im Normalfall sucht sich jede bzw. jeder Studierende nach eigenem Interesse und Neigung einen Lehrstuhl aus, an dem sie bzw. er die Masterarbeit schreiben möchte. Die meisten Lehrstühle veröffentlichen ihre angebotenen Themen. Oft werden Themen aber auch erst nach Absprache mit dem Studierenden gestellt, wobei Letzterer ein Vorschlagsrecht hat.

Die Anmeldung für die Masterarbeit erfolgt beim Prüfungsamt Angewandte Informatik. Für die Anmeldung ist das persönliche Erscheinen sowie die Vorlage des aktuellen Studentenausweises bzw. der Immatrikulationsbescheinigung notwendig.

# Ingenieurinformatik

<b>Design Optimization</b>					
<b>Modul-Nr:</b> 129007	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Design Optimization			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> JAVA-Grundkurs					
<p><b>Lernziele (learning outcomes):</b></p> <p>An important goal of this course is to present the theoretical foundations of mathematical optimization to students in a manner which allows them to use and employ design optimization for engineering applications in a sensible manner. This is achieved with a combination of theoretical lectures and practical exercises carried out using various computers software systems. In the second part of the course, students carry out team projects to solve engineering design tasks using their fundamental knowledge acquired in the first half of the course. Further, students must clearly present their projects results in a classroom setting to an audience with various technical background (the course is attended by students from applied computer science as well as by students from computational engineering). When the students have successfully complete this course, students</p> <ul style="list-style-type: none"> <li>• will be familiar with the types of numerical algorithms available today to solve, in particular, advanced engineering tasks;</li> <li>• will be able to program software components to carry out design optimization tasks or employ engineering software systems to include design optimization aspects;</li> <li>• will have a good understanding of the basics of design optimization to be able to select proper optimization techniques in a given engineering situation and be able to implement efficient numerical solutions.</li> </ul>					
<b>Inhalt:</b>					
<ul style="list-style-type: none"> <li>• Structural optimization as a tool for the optimal design of engineering tasks with respect to given quality objective functions, side constraints as well as inequality constraints.</li> </ul>					

## KAPITEL 3. INGENIEURINFORMATIK

- Development of optimization models for use in engineering applications
- Types of optimization categories (continuous, linear/non-linear, deterministic, simulation-based, multi-level, etc.)
- Strategies of optimization methods (classic indirect methods, direct numerical methods, global evolution strategies, partial swarm methods, distributed parallel methods, etc.)
- Software systems to implement design optimization tasks
- Application of design optimization paradigms to solve engineering tasks as a team project

**Lehrformen:**

lectures, exercises using computers, team projects

**Prüfungsformen:**

Semesterbegleitend; Completion of a team project (2-4 students); presentation of project results; oral examination

**Voraussetzungen für die Vergabe von Kreditpunkten:**

successful completion of a team project; presentation of project results in the classroom; oral examination

**Stellenwert der Note für die Endnote:** 6 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Markus König**

Prof. Dr. Markus König, Dr. Karlheinz Lehner

<b>Design sozio-technischer Informationssysteme</b>					
<b>Modul-Nr:</b> 260084	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung und Übung integriert			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> 4-8
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Mensch-Maschine-Interaktion					
<b>Lernziele (learning outcomes):</b> Die Teilnehmer/innen sind in der Lage, bei der Entwicklung von informationstechnischen Lösungen frühzeitig den späteren Anwendungskontext und die Nutzungsbedingungen zu berücksichtigen. Sie können helfen, die für die Systemeinführung und -Einsatz notwendigen organisatorischen Maßnahmen zu konzipieren.					
<b>Vermittelte Kompetenzen:</b>					
Kernkompetenzen:					
<ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> </ul>					
Fachspezifische Kompetenzen:					
<ul style="list-style-type: none"> <li>• Entwicklungsprozesse</li> <li>• Rechtliche Kompetenzen</li> </ul>					
<b>Inhalt:</b>					
Die Nutzung computerbasierter Kommunikations- und Informationssysteme wird in der Regel in menschliche Handlungsabläufe eingebettet. Die Vorlesung vermittelt Kenntnisse derjenigen Faktoren und Methoden, die bei der Entwicklung und Einführung informationstechnischer Systeme dazu beitragen, dass die Nutzung erfolgreich ist. Dabei wird davon ausgegangen, dass technische, organisatorische und soziale Strukturen integriert und angepasst werden müssen. Die Erfolgsfaktoren werden aus interdisziplinärer Sicht behandelt und anhand von Beispielen aus konkreten Anwendungsfällen erläutert. Dabei werden die folgenden Gebiete berücksichtigt:					
<ul style="list-style-type: none"> <li>• Arbeitswissenschaft und Ergonomie</li> <li>• Psychologie</li> <li>• Organisation und Management</li> <li>• Rechtliche Aspekte</li> <li>• Betriebswirtschaftliche Aspekte</li> <li>• Kommunikationstheorie</li> </ul>					

## KAPITEL 3. INGENIEURINFORMATIK

- Datenschutz

Es werden verschiedene Methoden vermittelt, die die Einführung von Informationssystemen unterstützen:

- Erhebung von Anforderungen und Ausgangsbedingungen
- Usability-Engineering
- Contextual Design
- Sozio-technische Modellierung
- Partizipation und Kommunikationsmoderation

### **Lehrformen:**

Abwechslung von Vorlesungsinhalten, Seminarbeiträgen der Studierenden, Präsenzübungen und Erörterung der Projektbearbeitung

### **Prüfungsformen:**

Semesterbegleitend; Seminarbeitrag, Präsentation der Bearbeitung einer Projektaufgabe in Kleingruppen, Prüfungsgespräch zur Überprüfung der Eigenleistung bei der Projektarbeit

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

Ein Seminarbeitrag (25% der Endnote) und Präsentation der Projektarbeit (75%)

**Stellenwert der Note für die Endnote:** 5 / 112

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

**Prof. Dr. Thomas Herrmann**  
Markus Jelonek Msc.

### **Sonstige Informationen:**

**Literatur:** Th. Herrmann (2012): Kreatives Prozessdesign. Springer-Gabler.

<b>IT im Engineering</b>					
<b>Modul-Nr:</b> 133640	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (2 SWS), Übung (2 SWS)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<p><b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• verfügen Studierende über ein breites, integriertes Wissen über die Funktionalitäten und Anwendungsspezifika prozessübergreifender server- oder cloud-basierter IT Anwendungen im Engineering bzw. im gesamten Produktlebenszyklus, um Digitalisierungsprojekte in Industrieunternehmen aus Sicht des Engineerings maßgeblich vorantreiben zu können.</li> <li>• kennen und verstehen Studierende wesentliche Methoden und Verfahren zur systematischen Erfassung, Darstellung und Analyse von Prozessen im Engineering und zur Konzeptionierung von unterstützenden IT-Anwendungen. Indem sie praktische Fallbeispiele und Aufgaben bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit entsprechenden Softwaresystemen auf konkrete Problemstellungen der Digitalisierung von Prozessen und der Informationsverarbeitung im Engineering übertragen.</li> <li>• haben Studierende ein umfassendes Verständnis vom Zusammenwirken der relevanten Softwaresysteme und von Integrationskonzepten aus Sicht des Engineerings erworben und können dieses situativ angepasst anwenden, um z.B. differenziert die Eignung von IT Verfahren in den Bereichen PDM/PLM, ERP, CRM, MES zu beurteilen.</li> <li>• verfügen Studierende über vertiefte, interdisziplinäre Methodenkompetenz und können Aufgabenstellungen der Einführung, des Betriebs und der Integration von prozessübergreifenden IT Anwendungen und industrieller Standard-Software reflektieren und bewerten sowie selbstgesteuert verfolgen.</li> <li>• können Studierende kooperativ Aufgabenstellungen der Prozessmodellierung, Konzeptentwicklung und Systemspezifikation von IT Anwendungen im Engineering in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen und über Sachverhalte umfassend kommunizieren.</li> </ul>					
<p><b>Inhalt:</b> Die Veranstaltung vermittelt eine Übersicht über Aufgaben und Funktionsweisen von prozessübergreifenden IT-Anwendungen im Produktlebenszyklus aus den Bereichen Produktentwicklung, Produktionsplanung, Fertigung und Montage, Service und Instandhaltung, Marketing und Vertrieb sowie Materialwirtschaft und Logistik, immer mit Bezug zu Aufgaben des Engineerings sowie den relevanten Informationsflüssen, Schnittstellen und Integrationsaspekten. Der Schwerpunkt liegt dabei auf dem erforderlichen Grundlagenwissen in Bezug auf Software und Informationstechnologien (z.B. Datenbanken, IoT Technologien, Middleware-</p>					

## KAPITEL 3. INGENIEURINFORMATIK

Konzepte sowie Cloud- und Edge Computing) und den relevanten methodischen Aspekten der Erstellung von Konzepten für die Prozessoptimierung, der Einführung von prozessunterstützenden IT-Lösungen sowie der IT-Organisation und des IT-Managements.

**Lehrformen:**

Vorlesung (2 SWS), Übung (2 SWS)

**Prüfungsformen:**

Klausur / 90 Minuten, Anteil der Modulnote : 100 Prozent. Schriftliche Klausur mit Fragen und Aufgaben zum gesamten Stoff der Vorlesung und der Übung

**Voraussetzungen für die Vergabe von Kreditpunkten:**

- Im Rahmen der Übung werden 5 Projektaufgaben gestellt, die bearbeitet und abgegeben werden müssen.
- Für die Zulassung zur Prüfung müssen mind. 4 Projektaufgaben positiv bewertet sein.

**Stellenwert der Note für die Endnote:** 6 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr.-Ing. Detlef Gerhard**

<b>Künstliche Intelligenz für Ingenieure</b>					
<b>Modul-Nr:</b> 141005	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Künstliche Intelligenz für Ingenieure (141005)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 94 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Diskrete Mathematik					
<b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• haben Studierende fachspezifisches Grundlagenwissen der symbolischen Informationsverarbeitung und deren Umsetzung in Algorithmen,</li> <li>• sind die Studierenden in der Lage, typische Problemstellungen der Wissensverarbeitung im ingenieurstechnischen Bereich zu erkennen und zu strukturieren,</li> <li>• haben Studierende erste Erfahrungen im Umgang mit Sprachen der künstlichen Intelligenz durch Übungen im CIP-Pool gesammelt,</li> <li>• können die Studierenden die Möglichkeiten und Grenzen der Künstlichen Intelligenz diskutieren.</li> </ul>					
<b>Inhalt:</b> Grundprinzipien der Wissensrepräsentation und der symbolischen Informationsverarbeitung mit Anwendungsbeispielen aus der Automatisierungstechnik, insbesondere Suchverfahren in gerichteten Graphen, regelbasierte Systeme, Aufbau und Funktionsweise logikbasierter Systeme, Anwendungen für die Fehlerdiagnose in technischen Systemen.					
<b>Lehrformen:</b> Vorlesung und Praxisübungen					
<b>Prüfungsformen:</b> Klausurarbeit – 120 Minuten					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Masterstudiengang Elektrotechnik und Informationstechnik					
<b>Stellenwert der Note für die Endnote:</b> 5 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Jan Lunze</b>					



**Sonstige Informationen:**

Letztmalig im SS21!

**Literatur:** Lunze, Jan "Künstliche Intelligenz für Ingenieure - Methoden zur Lösung ingenieurtechnischer Probleme mit Hilfe von Regeln, logischen Formeln und Bayesnetzen", Oldenbourg Wissenschaftsverlag, 2010

<b>Product Lifecycle Management</b>					
<b>Modul-Nr:</b> 138577	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Product Lifecycle Management (PLM)			<b>Kontaktzeit:</b> V: 2 SWS; Ü: 2	<b>Selbststudium:</b> 120 h	<b>Gruppengröße:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<p><b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• 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.</li> <li>• können Studierende prozessorientiert an PLM Aufgabenstellungen herangehen, diese reflektieren und bewerten sowie selbstgesteuert verfolgen.</li> <li>• können Studierende kooperativ PLM-Aufgabenstellungen in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen sowie über Sachverhalte umfassend kommunizieren.</li> </ul>					
<p><b>Inhalt:</b> Die Veranstaltung 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 Engineering und die Vorgehensweise bei der PLM-Einführung vermittelt.</p>					
<b>Lehrformen:</b>					

## KAPITEL 3. INGENIEURINFORMATIK

Vorlesungen, Übungen, Gruppenarbeiten, Moodle Unterstützung
<b>Prüfungsformen:</b> <ul style="list-style-type: none"><li>• Schriftliche Klausur mit Fragen und Aufgaben zum Stoff der Vorlesung und der Übung - Dauer: 90 Minuten, Anteil der Modulnote: 100%</li></ul>
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulklausur Prüfungsvorleistung: Im Rahmen der Übung werden 5 Projektaufgaben gestellt, die bearbeitet und abgegeben werden müssen. Für die Zulassung zur Modulklausur müssen mind. 4 Projektaufgaben positiv bewertet sein
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Studiengang Maschinenbau (Bachelor), Studiengang SEPM (Bachelor)
<b>Stellenwert der Note für die Endnote:</b> 6 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr.-Ing. Detlef Gerhard</b>

<b>Grundlagen der Verkehrsplanung und Verkehrstechnik</b>					
<b>Modul-Nr:</b> IV-8b	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Grundlagen der Verkehrsplanung und Verkehrstechnik			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Höhere Mathematik I und II					
<p><b>Lernziele (learning outcomes):</b> Das Modul soll die Studierenden mit den Grundlagen der Verkehrsplanung und der Straßenverkehrstechnik vertraut machen. Die Studierenden sollen lernen, Standardaufgaben aus diesem Bereich selbständig zu bearbeiten. Darüber hinaus sollen sie ein Grundverständnis für die verwendeten Methoden erlangen und Zusammenhänge innerhalb des Faches erkennen können. Sie sollen in die Lage versetzt werden, Vorgänge und Lösungen aus der Verkehrsplanung und Verkehrstechnik kritisch zu beurteilen.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• kennen Studierende einschlägige Methoden zur Erhebung und Analyse von Verkehrsdaten,</li> <li>• verstehen Studierende, wie das künftige Verkehrsgeschehen in einem Straßennetz anhand von Verkehrsdaten prognostiziert werden kann,</li> <li>• kennen Studierende elementare theoretische Grundlagen von Bewertungsmodellen und Simulationsverfahren in der Verkehrsplanung und Verkehrstechnik,</li> <li>• können die Studierenden Strecken von Autobahnen und Landstraßen sowie vorfahrtge-regelte und signalgesteuerte Knotenpunkte verkehrstechnisch bemessen und bewerten,</li> <li>• können die Studierenden die gängigen Richtlinien zur verkehrstechnischen Bemessung und zur Wirtschaftlichkeitsanalyse von Straßenverkehrsanlagen anwenden.</li> </ul>					
<p><b>Inhalt:</b> Die Lehrveranstaltung behandelt das Basiswissen der Verkehrsplanung und der Straßenverkehrstechnik. Hierzu gehören:</p> <ul style="list-style-type: none"> <li>• Verkehrsanalyse (Erhebungs- und Zählmethoden)</li> <li>• <b>4-Stufen-Algorithmus der klassischen Verkehrsplanung:</b> <ul style="list-style-type: none"> <li>– Verkehrserzeugungsmodelle und Prognoseverfahren</li> <li>– Verkehrsverteilung</li> <li>– Verkehrsaufteilung auf verschiedene Verkehrssysteme</li> <li>– Verkehrsumlegung auf die Strecken eines Netzes</li> </ul> </li> </ul>					

## KAPITEL 3. INGENIEURINFORMATIK

- Kinematische Grundlagen der Verkehrstechnik
- Statistische Grundbegriffe, Warteschlangentheorie
- Verkehrsfluss auf Straßen, Fundamentaldiagramm
- Strecken von Autobahnen und Landstraßen
- Vorfahrtgeregelte Knotenpunkte
- Knotenpunkte mit Lichtsignalanlage
- Verkehrssicherheit
- Verkehrslärm
- Verfahren der Wirtschaftlichkeitsrechnung für die Infrastrukturplanung

**Lehrformen:**

Vorlesungen (2 SWS) und Übungen (2 SWS)

**Prüfungsformen:**

Modulklausur (90 min), optionale Hausarbeit zur Erlangung von Bonuspunkten für die Modulklausur

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Verwendung des Moduls (in anderen Studiengängen):** Pflichtmodul im Bachelor-Studiengang Umwelttechnik und Ressourcenmanagement sowie Teil eines Pflichtmoduls im Bachelor-Studiengang Bauingenieurwesen

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr.-Ing. Justin Geistefeldt**

**Sonstige Informationen:****Literatur:**

1. Schnabel, Lohse: Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung, 3. Auflage, Beuth-Verlag
2. Steierwald, Künne, Vogt (Hrsg.): Stadtverkehrsplanung, 2. Auflage, Springer-Verlag
3. Köhler (Hrsg.): Verkehr – Straße, Schiene, Luft. Verlag Ernst und Sohn
4. Einschlägige Richtlinien und Merkblätter

<b>Verkehrstechnik</b>					
<b>Modul-Nr:</b> MISSING	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes SoSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>			<b>Kontaktzeit:</b>	<b>Selbststudium:</b>	<b>Gruppen- größe:</b>
a) Vorlesung Verkehrssteuerung			a) 2 SWS	a) 60h	beliebig
b) Vorlesung Modellierung und Simulation des Verkehrsflusses			b) 2 SWS	b) 60h	
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Kenntnisse in Verkehrstechnik					
<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 Koordination 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, vorfahrtgeregeltelte Knotenpunkte, Festzeitsteuerung von Signalanlagen, Grüne Welle, Koordinierung im Netz, verkehrsabhängige Steuerung einschließlich Signalprogrammgebung, 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, Fahrzeugfolgetheorie, mikroskopische Verkehrsflusssimulation.					
<b>Lehrformen:</b> Powerpoint-Präsentationen, Tafel, Vorführungen und Übungen am PC					

### KAPITEL 3. INGENIEURINFORMATIK

<b>Prüfungsformen:</b> Abschlussprüfung; Klausur / 120 Minuten; Klausur über das gesamte Modul
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. Bauingenieurwesen
<b>Stellenwert der Note für die Endnote:</b> 6 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr.-Ing. Justin Geistefeldt</b>  a) Apl. Prof. Dr.-Ing. Ning Wu  b) Prof. Dr.-Ing. Justin Geistefeldt
<b>Sonstige Informationen:</b> <b>Literatur:</b> Ausführliches Skript zur Lehrveranstaltung; Schnabel, Lohse: Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung, Band 1- Verkehrstechnik, Beuth Verlag Einschlägige Richtlinien und Merkblätter (werden in der Vorlesung genannt)

<b>Wissensbasierte Methoden</b>					
<b>Modul-Nr:</b> 128701	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 2/4. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Wissensbasierte Methoden			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Objektorientierte Modellierung					
<p><b>Lernziele (learning outcomes):</b> Neben den rein numerischen Berechnungsverfahren im Ingenieurwesen gibt es auch vielfältige Algorithmen, die nicht-numerischer Art sind und stattdessen mit symbolischen Größen Beziehungen zwischen den Komponenten eines Systems festhalten. Ein Verständnis solcher Verfahren soll geschult werden und den Studierenden befähigen solche Verfahren auch anzuwenden. Zu den behandelten Themen gehören neben KI-spezifisches Wissen wie Grundlagen zu Neuronalen Netzen oder zur Fuzzy Logik auch Kompetenzen wie analytisches und logisches Denken, fächerübergreifendes Denken und Arbeiten sowie die Fähigkeit zur Teamarbeit und Teamfähigkeit. Der Einsatz aktueller Hard- und Software zur Lösung von KI Problemen aus dem Bereich des Ingenieurwesens gibt einen Einblick in das breite Spektrum an Lösungsansätzen der KI. Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• kennen Studierende die Grundlagen verschiedener KI-Techniken wie Neuronale Netze oder der Fuzzy Logik bzw. die theoretischen Grundlagen der Aussage- und Prädikatenlogik;</li> <li>• verstehen Studierende durch Bearbeitung eines Team-Projekts aus dem Bereich der KI, wie sowohl die KI-Theorie als auch der Einsatz von KI-Software zur Lösung von Ingenieur Anwendungen erfolgreich eingesetzt werden kann.</li> </ul>					
<p><b>Inhalt:</b></p> <ul style="list-style-type: none"> <li>• Historische Entwicklung und Bedeutung der wissensbasierten Methodik für moderne Entwurfs-, Planungs-, Konstruktions- und Fertigungsprobleme im Ingenieurwesen</li> <li>• Grundlegende Techniken der Repräsentation, Formalisierung und Verarbeitung von Ingenieurwissen (Aussagenlogik, Prädikatenlogik, Regelbasierte Systeme)</li> <li>• Methoden und Anwendungen der Fuzzy-Logik zur Erfassung unscharfer Sachverhalte im Ingenieurwesen und zur Modellierung komplexer technischer Systeme</li> <li>• AI-orientierte Programmiersprachen (Prolog, Jess)</li> <li>• Neuronale Netze</li> <li>• Integriertes Projekt aus dem Bereich der Fuzzy-Logik</li> </ul>					



## KAPITEL 3. INGENIEURINFORMATIK

**Lehrformen:**

- Vorlesungen
- Übungen am Rechner und im Seminarraum
- Gruppenarbeiten

**Prüfungsformen:**

Semesterbegleitend; - Bearbeitung eines Projekts im Team (2-4 Studierende) - Präsentation der Projekt-Ergebnisse - Abgabegespräch

**Voraussetzungen für die Vergabe von Kreditpunkten:**

- Erfolgreiche Bearbeitung eines Projekts im Team
- Präsentation der Ergebnisse im Kurzvortrag
- Abgabegespräch

**Stellenwert der Note für die Endnote:** 6 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Markus König**  
Prof. Dr. Markus König, Dr. Karlheinz Lehner

<b>Grundlagen der automatischen Spracherkennung</b>					
<b>Modul-Nr:</b> 141044	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung und Praxisübungen			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine  <b>Empfohlene Vorkenntnisse:</b></p> <ul style="list-style-type: none"> <li>• Grundkenntnisse der digitalen Signalverarbeitung und der Wahrscheinlichkeitsrechnung</li> <li>• Grundlegende Programmierkenntnisse</li> </ul>					
<p><b>Lernziele (learning outcomes):</b>            Die Teilnehmer verstehen die theoretischen und praktischen Grundlagen automatischer Spracherkennungssysteme. Sie sind in der Lage, die Kernalgorithmen eines einfachen Spracherkenners selbstständig zu implementieren und verstehen die Prinzipien von aktuellen Erkennungssystemen für kleines und großes Vokabular. Dabei wird auch ein Verständnis für die Entwicklung von automatischen Mustererkennungsverfahren für ein breites Anwendungsfeld entwickelt.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Kritikfähigkeit</li> <li>• Selbständiges Lernen und Arbeiten</li> <li>• Teamarbeit und Teamfähigkeit</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Algorithmik</li> <li>• Künstliche Intelligenz und Maschinelles Lernen</li> <li>• Mathematik</li> <li>• Bild- und Signalverarbeitung</li> <li>• Programmieren</li> </ul>					

## KAPITEL 3. INGENIEURINFORMATIK

Technologien:

- Python

### **Inhalt:**

Die Vorlesung vermittelt Grundlagen und Algorithmen der maschinellen Spracherkennung in der Form, in der sie in aktuellen Systemen zur Erkennung fließender Sprache eingesetzt werden. Die folgenden Themen werden behandelt:

- Grundlagen: Phonetik, Sprachwahrnehmung
  - Statistische Methoden: Klassifikation, Schätztheorie
  - Klassifikation mittels Deep Neural Networks
  - Merkmalsextraktion: Merkmale im Zeit- und Frequenzbereich, Cepstralanalyse
  - Spracherkennung mit Hidden Markov Modellen: Algorithmen, Modellinitialisierung, Baum-Welch-Reestimation, Numerische Aspekte, Systeme zur Einzel- und Verbundwörtererkennung, HMM/DNN-Systeme
- Gleichzeitig werden in einem Programmierpraktikum die eingeführten Methoden angewandt.

Die Übung ist projektorientiert; alle Übungsaufgaben zusammengenommen ergeben einen Verbundwörterkennner für fließend gesprochene Ziffernketten. Dieser wird in Arbeitsgruppen von 2-3 Studenten erarbeitet.

### **Lehrformen:**

Vorlesung und Praxisübungen

### **Prüfungsformen:**

Klausur schriftlich, 120 Minuten

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Stellenwert der Note für die Endnote:** 6 / 112

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

**Prof. Dr.-Ing. Dorothea Kolossa**

# Programmier- und Simulationstechnik

<b>Fundamentals of GPU Programming</b>					
<b>Modul-Nr:</b> 141374	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> C (programming language)					
<b>Lernziele (learning outcomes):</b> To learn how to program on graphics processing units (GPUs).					
<b>Vermittelte Kompetenzen:</b>					
Kernkompetenzen:					
<ul style="list-style-type: none"> <li>• Selbständiges Lernen und Arbeiten</li> </ul>					
Fachspezifische Kompetenzen:					
<ul style="list-style-type: none"> <li>• Software-Entwurfsmethoden</li> </ul>					
Technologien:					
<ul style="list-style-type: none"> <li>• CUDA</li> <li>• C</li> <li>• Matlab</li> </ul>					
<b>Inhalt:</b>					
<ol style="list-style-type: none"> <li>1. GPU as a modern means for general-purpose massively parallel computations</li> <li>2. General GPU architecture and CUDA operational model</li> <li>3. Basic CUDA syntax</li> </ol>					

## KAPITEL 4. PROGRAMMIER- UND SIMULATIONSTECHNIK

4. Optimization strategies in GPU programming
5. Case study of general-purpose GPU programming

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

Semesterbegleitend; Projekt

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandendes Projekt

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Dr. Denis Eremin**

<b>Simulationstechnik</b>					
<b>Modul-Nr:</b> 127012	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Objektorientierte Modellierung					
<p><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.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Projekt- und Zeitmanagement</li> <li>• Teamarbeit und Teamfähigkeit</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Simulationsmethoden</li> <li>• Optimierungsmethoden</li> <li>• Programmieren</li> </ul> <p>Technologien:</p> <ul style="list-style-type: none"> <li>• AnyLogic</li> <li>• Java</li> </ul>					
<p><b>Inhalt:</b> Es werden Vorgehensweisen zur Simulation komplexer Systeme vermittelt. Folgende Themen werden behandelt:</p> <ul style="list-style-type: none"> <li>• Systemanalyse und Modellbildung</li> <li>• System Dynamics</li> <li>• Ereignisdiskrete Simulation</li> </ul>					

## KAPITEL 4. PROGRAMMIER- UND SIMULATIONSTECHNIK

- Multiagentensimulation
- Aufbereitung von Eingangsdaten
- Stochastische Simulation
- Simulationsgestützte Optimierung
- Einführung in die Software AnyLogic

Innerhalb der Projektarbeit werden aktuelle Fragestellungen aus den Bau- und Umweltingenieurwissenschaften (Montage- und Logistikprozesse, Fußgängersimulation, Schadstoffausbreitung, etc.) aufgearbeitet und mit Hilfe einer objekt-orientierten Simulationssoftware analysiert. Den Studierenden werden Softwarelizenzen durch den Lehrstuhl zur Verfügung gestellt. Die Programmierung erfolgt unter Verwendung der Programmiersprache Java.

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

Semesterbegleitend; Hausarbeit/Projektarbeit

**Voraussetzungen für die Vergabe von Kreditpunkten:**

erfolgreiche Projektarbeit

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Markus König**

**Sonstige Informationen:****Literatur:**

1. Banks, J.; Carson II, J. S.; Nelson, B. L.; Nicol, D. M. (2005): Discrete-Event System Simulation, Pearson Prentice Hall
2. Bossel, H. (1994): Modellbildung und Simulation : Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme, ein Lehr- und Arbeitsbuch, Vieweg Verlag
3. Biethahn, J.; Lackner, A.; Range, M; Brodersen, O. (2004): Optimierung und Simulation, Oldenbourg Verlag, München
4. Simulationssoftware AnyLogic der Firma XJ Technologies, <http://www.anylogic.com>

<b>3D-Simulation in der Automatisierungstechnik</b>					
<b>Modul-Nr:</b> 139050	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1-4. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) 3D-Simulation in der Automatisierungstechnik			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppengröße:</b> 20 Studierende (Übungen)
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Besuch der Vorlesung "Grundlagen der Automatisierungstechnik"					
<p><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.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <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>					
<p><b>Inhalt:</b>                      Die Vorlesung deckt die folgenden Themenbereiche ab</p> <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> <p>Im Bereich der Kompetenzen im Kontext Digitalisierung ist besonders die Übung zu nennen, diese besteht aus der praktischen Umsetzung der von Modellierungs-, Programmierungs-</p>					



## KAPITEL 4. PROGRAMMIER- UND SIMULATIONSTECHNIK

und Simulationsaufgaben mit einem kommerziell verfügbaren und industriell eingesetzten 3D-Robotersimulations- und Offline-Programmiersystem.
<b>Lehrformen:</b> Gemeinsame Vorlesung und Übung in kleineren Gruppen
<b>Prüfungsformen:</b> Klausur (90 Minuten)
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Masterstudiengang Maschinenbau
<b>Stellenwert der Note für die Endnote:</b> 6 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Dr.-Ing. Alfred Hypki
<b>Sonstige Informationen:</b> Bei geringer Teilnehmerzahl kann die Prüfung auch mündlich (30 min.) angeboten werden.

<b>Complexity Economics and Agent-based Modeling</b>					
<b>Modul-Nr:</b> 075240	<b>Credits:</b> 10 CP	<b>Workload:</b> 300 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung  b) Übung (075246)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 240 h	<b>Gruppen- größe:</b> 30 Studieren- de aus den Wirtschafts- wissenschaften und der An- gewandten Informatik
<p><b>Teilnahmevoraussetzungen:</b> Gute Englischkenntnisse Grundlegende Programmierkenntnisse</p> <p><b>Empfohlene Vorkenntnisse:</b> Successful completion of a “Studienleistung”. Macroeconomics I and Macroeconomics II strongly recommended. Willingness to learn computer programming.</p>					
<p><b>Lernziele (learning outcomes):</b> Das Modul verfolgt das Ziel, Studierende der Angewandten Informatik sowie der Wirtschaftswissenschaft in die Methode der agentenbasierten Modellierung und den dazugehörigen Simulationstechniken einzuführen. Hierbei lernen Studierende wie volkswirtschaftliche und sozialwissenschaftliche Fragestellungen entworfen, implementiert und analysiert werden. Im Rahmen der Seminararbeit sollen die Studierenden ein eigenes Modell implementieren und analysieren.</p> <p><b>Im Rahmen des Moduls erwerben Studierende folgende Kenntnisse:</b></p> <ol style="list-style-type: none"> <li>1) Grundlegende Kenntnisse der Komplexitätsökonomik und agentenbasierter Modellierung</li> <li>2) Überführung von wissenschaftlichen Fragestellungen aus dem Bereich Volkswirtschaftslehre in agentenbasierte Computersimulationen</li> <li>3) Praktische Arbeit mit agentenbasierten Modellen, Interpretation von Simulationsergebnissen und mögliche Grenzen des Ansatzes</li> <li>4) Grundlegende Kenntnisse, um eigene Modelle zu implementieren und Simulationen selbstständig durchzuführen</li> <li>5) Wissenschaftliches Schreiben</li> <li>6) Die Programmiersprache NetLogo</li> </ol>					
<p><b>Inhalt:</b> Komplexitätsökonomik ist eine neue Denkschule im Bereich der Volkswirtschaftslehre, der zunehmend an Relevanz gewinnt. Hierbei wird eine Volkswirtschaft als ein komplexes, adaptives System betrachtet, das sich aus verschiedenen Agenten, z.B. Firmen, Haushalten</p>					

## KAPITEL 4. PROGRAMMIER- UND SIMULATIONSTECHNIK

und Banken, zusammensetzt. Im Vergleich zu bisherigen volkswirtschaftlichen Ansätzen sind die Agenten des Systems nicht (vollständig) rational und das System entwickelt sich aus den Interaktionen zwischen den (heterogenen) Agenten. Komplexitätsökonomik eignet sich sehr zur Untersuchung von sozialen Transformations- und Innovationsprozessen, wie der Digitalisierung.

In der Vorlesung wird Komplexitätsökonomik und die Methode der agentenbasierten Computersimulation vorgestellt und gezeigt, wie sie zur Analyse komplexer ökonomischer Systeme angewendet werden können. Im Rahmen der Übung werden diese Kenntnisse vertieft. Zudem wird geübt, wie agentenbasierte Modelle mithilfe NetLogo implementiert und simuliert werden.

**Lehrformen:**

Vorlesung und Übung

**Prüfungsformen:**

Semesterbegleitend; Seminararbeit, Klausur

**Voraussetzungen für die Vergabe von Kreditpunkten:****Bestehen der folgenden Leistungen:**

- 1) Studienleistung (unbenotet)
- 2) Seminararbeit und Klausur (benotet)

**Verwendung des Moduls (in anderen Studiengängen):** MSc Management and Economics, MSc Economic Policy Consulting

**Stellenwert der Note für die Endnote:** 10 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Michael Roos**

**Sonstige Informationen:**

Verbindliche Anmeldung in der ersten Sitzung und durch Registrierung bei Moodle. Studierende der angewandten Informatik müssen sich beim Zentrum für ökonomische Bildung (ZfÖB) zum Modul anmelden, um eine Prüfungsleistung erbringen zu können.

**Literatur:** Will be announced at the beginning of the module.

<b>High-Performance Computing on Clusters</b>					
<b>Modul-Nr:</b> N/A	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes WiSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung High-Performance Computing on Clusters  b) Übung			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Lernziele (learning outcomes):</b> In this module, the students acquire professional skills to program and employ parallel computing clusters. Theoretical properties of distributed-memory systems and programming patterns are conveyed as well as the practical implementation.					
<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, computer lab, numerical experiments					
<b>Prüfungsformen:</b> Abschlussprüfung; Klausur / 120 Minuten					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. Bauingenieurwesen, M.Sc. Computational Engineering					
<b>Stellenwert der Note für die Endnote:</b> 6 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Jun. Prof. Dr. Andreas Vogel</b>					

**Sonstige Informationen:**

**Literatur:**

- W. Hackbusch, Iterative Solution of Large Sparse Systems of Equations, Springer, 1994
- Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003
- MPI Application Programming Interface (2015), [www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf](http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf)
- W. Gropp, E. Lusk, A. Skjellum, Using MPI, MIT Press, 2014
- S. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, MPI – The Complete Reference, MIT Press, 1998
- T. Rauber, G. Rünger, Parallel Programming: for Multicore and Cluster Systems, Springer, 2013
- C. Douglas, G. Haase, U. Langer, A Tutorial on Elliptic PDE Solvers and Their Parallelization, SIAM, 2003
- additional literature will be announced in the lecture

<b>High-Performance Computing on Multi- and Manycore Processors</b>					
<b>Modul-Nr:</b> N/A	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes SoSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> a) Vorlesung High-Performance Computing on Multi- and Manycore Processors b) Übung			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Lernziele (learning outcomes):</b> In this module, the students acquire professional skills to program multi- and manycore processors employing multi-threaded execution and handling shared-memory access patterns. Theoretical properties are conveyed as well as practical implementation. Via presentations of selected topics, students attain the ability to survey and acquire knowledge on advanced scientific topics independently and are qualified to illustrate such topics in the form of a presentation and numerical examples.					
<b>Inhalt:</b> The lecture addresses parallelization for multi- and manycore processors. Thread-based programming concepts (pthreads, C++11 threads, OpenMP, OpenCL) are introduced and best-practice implementation aspects are highlighted based on applications from scientific computing. In the first part, the lecture provides an overview on relevant data structures, solver techniques and programming patterns from scientific computing. An introduction to multi- threading programming on multicore systems is then provided with special attention to shared-memory aspects. Parallelization patterns are discussed and highlighted. Numerical experiments and self-developed software implementations are used to discuss and illustrate the presented content. In the second part, students are assigned advanced topics for shared-memory computation from the engineering science including finite element methods and artificial intelligence. Based on a scientific paper, students present their topic to the lecture audience in form of a beamer presentation and numerical illustrations.					
<b>Lehrformen:</b> Beamer, computer lab, numerical experiments					
<b>Prüfungsformen:</b> Semesterbegleitend; Hausarbeit					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> mit mindestens ausreichend bewertete Hausarbeit					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. Bauingenieurwesen, M.Sc. Computational Engineering					

<b>Stellenwert der Note für die Endnote: 6 / 112</b>
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Jun. Prof. Dr. Andreas Vogel</b>
<b>Sonstige Informationen:</b> <b>Literatur:</b> <ul style="list-style-type: none"><li>• G. Hager, G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010</li><li>• T. Rauber, G. Rünger, Parallel Programming: for Multicore and Cluster Systems, Springer, 2013</li><li>• OpenMP Application Programming Interface (2015), <a href="http://www.openmp.org/wp-content/uploads/openmp4.5.pdf">www.openmp.org/wp-content/uploads/openmp4.5.pdf</a></li><li>• OpenCL Application Programming Interface (2014), <a href="http://www.khronos.org/registry/OpenCL/specs/1.2.pdf">www.khronos.org/registry/OpenCL/specs/1.2.pdf</a></li><li>• additional literature will be announced in the lecture</li></ul>

<b>Knowledge Graphs</b>					
<b>Modul-Nr:</b> 150336	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> SoSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung Knowledge Graphs  b) Übungen			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Basic knowledge about the following topics is highly recommended but not mandatory: Graph theory, set theory, databases, logic.</p>					
<p><b>Lernziele (learning outcomes):</b> In this lecture, students will learn about the foundations of modelling, querying, publishing, and reasoning over KGs. The topics will be complemented with exercises and Jupyter Notebooks (<a href="https://jupyter.org/">https://jupyter.org/</a>) to show how KG technologies work in practice.</p>					
<p><b>Inhalt:</b> Knowledge Graphs (KG) allow for representing inter-connected facts or statements annotated with semantics. In KGs, concepts and entities are typically modeled as nodes while their connections are modeled as directed and labeled edges, creating a graph. In recent years, KGs have become core components of modern data ecosystems. KGs, as building blocks of many Artificial Intelligence approaches, allow for harnessing and uncovering patterns from the data. Currently, KGs are used in the data-driven business processes of multinational companies like Google, Microsoft, IBM, eBay, and Facebook. Furthermore, thousands of KGs are openly available on the web following the Linked Data (<a href="https://lod-cloud.net/">https://lod-cloud.net/</a>) principles. The specific topics covered in the lecture are as follows:</p> <ol style="list-style-type: none"> <li>1. Introduction to Knowledge Graphs</li> <li>2. The Resource Description Framework (RDF)</li> <li>3. RDF Schema (RDFS)</li> <li>4. The SPARQL Query Language</li> <li>5. Semantics of SPARQL</li> <li>6. Linked Data: Knowledge Graphs and Ontologies on the Web</li> <li>7. The Web Ontology Language (OWL)</li> <li>8. Entailment Regimes</li> <li>9. Reasoning over Knowledge Graphs</li> </ol>					



## KAPITEL 4. PROGRAMMIER- UND SIMULATIONSTECHNIK

10. Property Graphs
11. Knowledge Graph Applications
<b>Lehrformen:</b> Vorlesung mit begleitenden Übungen
<b>Prüfungsformen:</b> Written exam
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Studiengang Mathematik
<b>Stellenwert der Note für die Endnote:</b> 5 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Jun.-Prof. Maribel Acosta Deibe</b>
<b>Sonstige Informationen:</b> <b>Literatur:</b> <ul style="list-style-type: none"><li>• Aidan Hogan et al. Knowledge Graphs. 2020. <a href="https://arxiv.org/pdf/2003.02320.pdf">https://arxiv.org/pdf/2003.02320.pdf</a></li><li>• Andreas Harth. Introduction to Linked Data. (Specific chapters will be provided in the lecture).</li><li>• Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph. Foundations of Semantic Web Technologies. Chapman and Hall/CRC, 2009.</li></ul>

# Neuroinformatik

<b>Autonomous Robotics: Action, Perception and Cognition</b>					
<b>Modul-Nr:</b> 310501 (Übung: 310511)	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 135 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<p><b>Lernziele (learning outcomes):</b>            We review experimental results in movement science, discuss mathematical models of movement generation, and use robotic instantiations of such models to illustrate their function. The mathematical language that pervades the theoretical work reviewed in the course comes from the theory of dynamical systems. The course includes tutorials on basic concepts in dynamical systems theory. The exercises provide opportunities to use those concepts in a variety of contexts.</p> <p>Another goal of the course is to expose students to interdisciplinary science. The exercises include readings of review papers in different relevant fields. An essay exercise practices reading and writing at the level of academic research papers.</p> <p>The course consists of a weekly 2-hour lecture, followed by a 1-hour exercise session. Exercise sheets given out each week must be handed in and individually corrected. They are discussed in the week after they are due.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Literaturrecherche und Dokumentation</li> <li>• Selbständiges Lernen und Arbeiten</li> <li>• Wissenschaftliches Arbeiten und Schreiben</li> </ul> <p>Fachspezifische Kompetenzen:</p>					

## KAPITEL 5. NEUROINFORMATIK

- Mathematik
- Künstliche Intelligenz und Maschinelles Lernen

**Inhalt:**

Humans are the dexterous species. We excel at movement generation, in particular, at handling objects and generating the complex sequences of actions that achieve goals. This course looks at the fundamental processes of movement generation in humans and other animals and characterizes the special properties of human movement that emerge from the neural foundation. Object-oriented movement generation entails not only the timing and control of movement, but also object perception, scene representation, and the organization and planning of sequences. Movement generation thus cuts across a wide range of neural processes.

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

Klausur, Übungsaufgaben, Hausarbeit/Projektarbeit

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Stellenwert der Note für die Endnote:** 6 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Gregor Schöner**

**Sonstige Informationen:**

Eine Prüfung in 'Movement generation by Humans and Robots: A dynamical systems perspective' ist nicht möglich, wenn das ausgelaufene Modul 'Autonomous Robotics: Action, Perception, and Cognition' bereits erfolgreich bestanden wurde.

**Literatur:**

1. Valentino Braitenberg: Vehicles. Experiments in Synthetic Psychology, MIT Press, Cambridge, Mass 1984
2. Gregor Schöner, Michael Dose, Christoph Engels: Dynamics of behavior: Theory and applications for autonomous robotic architectures. Robotics and Autonomous Systems, 16:213-245 (1995)
3. Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis, and Gregor Schöner: Dynamic Neural Fields as Building Blocks of a Cortex-Inspired Architecture for Robotic Scene Representation. IEEE Transactions Autonomous Mental Development 3:74-91 (2011)

<b>Computational Neuroscience: Neural Dynamics</b>					
<b>Modul-Nr:</b> 310001 (Übung: 310011)	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 135 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Keine					
<b>Lernziele (learning outcomes):</b> <ul style="list-style-type: none"> <li>• Gain experience in interdisciplinarity bridging computer science and cognitive science.</li> <li>• Learn the concepts and methods of nonlinear dynamical systems in a concrete applied context.</li> <li>• Improve familiarity with methods of quantitative natural science, including measurement, graphing observables as a function of experimental control parameters and using models to interpret data.</li> <li>• Read scientific literature.</li> </ul>					
<b>Vermittelte Kompetenzen:</b>					
Kernkompetenzen:					
<ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Literaturrecherche und Dokumentation</li> <li>• Selbständiges Lernen und Arbeiten</li> <li>• Wissenschaftliches Arbeiten und Schreiben</li> </ul>					
Fachspezifische Kompetenzen:					
<ul style="list-style-type: none"> <li>• Mathematik</li> <li>• Künstliche Intelligenz und Maschinelles Lernen</li> </ul>					
<b>Inhalt:</b>					
<p>This course provides an introduction into the theoretical cognitive and functional neurosciences from a particular theoretical vantage point, the dynamical systems approach. This approach emphasizes the evolution in time of behavioral and neural patterns as the basis of their analysis and synthesis. Dynamic stability, a concept shared with the classical biological</p>					

## KAPITEL 5. NEUROINFORMATIK

cybernetics framework, is one cornerstone of the approach. Instabilities (or bifurcations) extend this framework and provide a basis for understanding flexibility, task specific adjustment, adaptation, and learning. The course includes tutorial modules that provide mathematical foundations. Theoretical concepts are expounded in reference to a number of experimental model systems which include the coordination of movement, postural stability, the perception of motion, and elementary forms of embodied cognition. In the spirit of Braitenberg's "synthetic psychology", autonomous robots are used to illustrate some of the ideas.

Exercises are integrated into the lectures. They consist of elementary mathematical exercises, the design of (thought) experiments and their analysis, and the design of simple artificial systems, all on the basis of the theoretical framework exposed in the main lectures.

One exercise takes the form of an essay for which participants read a scientific paper and answer questions in a longer illustrated text.

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

Klausur, Übungsaufgaben, Hausarbeit/Projektarbeit

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Stellenwert der Note für die Endnote:** 6 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Gregor Schöner**

**Sonstige Informationen:****Literatur:**

1. Martin Braun: Differential equations and their applications, Springer Verlag, New York, 1993
2. Gregor Schöner and Scott Kelso: Dynamic Pattern Generation in Behavioral and Neural Systems. Science 239: 1513-1520 (1988)
3. Gregor Schöner: Dynamical Systems Approaches to Cognition. In: The Cambridge Handbook of Computational Psychology,
4. Ron Sun, (ed.), Cambridge University Press (2008), pages 101-126

<b>Computational Neuroscience: Vision and Memory</b>					
<b>Modul-Nr:</b> 310504 (Übung: 310514)	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Computational Neuroscience: Vision and Memory			<b>Kontaktzeit:</b> 6 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> 20
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> The mathematical level of the course is mixed but generally high. The tutorial is almost entirely mathematical. Mathematics required include calculus (functions, derivatives, integrals, differential equations, ...), linear algebra (vectors, matrices, inner product, orthogonal vectors, basis systems, ...), and a bit of probability theory (probabilities, probability densities, Bayes' theorem, ...).</p>					
<p><b>Lernziele (learning outcomes):</b> After the successful completion of this course the students:</p> <ul style="list-style-type: none"> <li>• know basic neurobiological facts about the visual system and the hippocampus,</li> <li>• know a number of related models and methods in computational neuroscience,</li> <li>• understand the mathematics of these methods,</li> <li>• can communicate about all this in English.</li> </ul>					
<p><b>Inhalt:</b> This lecture covers basic neurobiology and models of selforganization in neural systems, in particular addressing vision (receptive fields, neural maps), and hippocampus (navigation and associative memory).</p>					
<p><b>Lehrformen:</b> Flipped/inverted classroom</p>					
<p><b>Prüfungsformen:</b> Oral exam, ca 20 min</p>					
<p><b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur</p>					
<p><b>Stellenwert der Note für die Endnote:</b> 6 / 112</p>					
<p><b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Laurenz Wiskott</b></p>					
<p><b>Sonstige Informationen:</b> There is a lecture, which provides the content, and a tutorial, where you solve exercises and</p>					

## KAPITEL 5. NEUROINFORMATIK

can deepen your understanding of the content. The exercises are solved in the tutorial in a group effort, not at home, which is the reason why it takes 3 hours rather than the usual 1.5 hours.

<b>Machine Learning: Evolutionary Algorithms</b>					
<b>Modul-Nr:</b> 310008 (Übung: 310018)	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung mit Integrierter Übung (flipped classroom)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> The course is designed for Master students of the Angewandte Informatik program. The lecture Mathematics for Modeling and Data Analysis is not mandatory but recommended as a background.</p> <p>Participants should be familiar with linear algebra and elementary probability theory. For example, the following terms should be well known:</p> <ul style="list-style-type: none"> <li>• vector, basis, linear map, matrix</li> <li>• norm, inner product, orthogonal</li> <li>• probability, distribution, density, quantile</li> <li>• normal distribution, expectation, variance, covariance</li> </ul> <p>Several concepts from higher mathematics will appear during the course. These will be introduced on the fly.</p>					
<p><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.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• kennen die Teilnehmer die wichtigsten Klassen direkter Optimierungsverfahren und ihre algorithmischen Komponenten,</li> <li>• haben die Teilnehmer ein tiefes Verständnis evolutionärer Algorithmen, insbesondere für kontinuierliche Probleme,</li> <li>• kennen die Teilnehmer eine Reihe spezifischer Problemschwierigkeiten und die zugehörigen algorithmischen Komponenten, welche diese Adressieren,</li> <li>• können die Teilnehmer elementare Laufzeitanalysen durchführen und verstehen die wichtigsten Konvergenzklassen</li> <li>• können die Teilnehmer Optimierungsverfahren selbst implementieren und zur Lösung neuer Probleme anwenden.</li> </ul>					



## KAPITEL 5. NEUROINFORMATIK

<b>Inhalt:</b> Breiter Überblick über Optimierungsverfahren. Evolutionäre Optimierungsverfahren für black-box Optimierungsverfahren Algorithmische Komponenten für schlechte Konditionierung, Multimodalität, Rauschen, Nebenbedingungen und Mehrzieloptimierung. Konvergenz- und Laufzeitanalyse.
<b>Lehrformen:</b> flipped classroom
<b>Prüfungsformen:</b> Klausur 90 Minuten, Wiederholungsklausur nach 6 Monaten
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Master Medizinphysik Master Physik Sämtliche Ingenieurstudiengänge Die Veranstaltung wird auch regelmäßig von Studierenden sämtlicher Ingenieurstudiengänge, der Chemie, der Mathematik, aus cognitive science, von Doktoranden, sowie von Studierenden der TU Dortmund (insbesondere automation and robotics) nachgefragt.
<b>Stellenwert der Note für die Endnote:</b> 6 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Jun.-Prof. Dr. Tobias Glasmachers

<b>Master-Praktikum: Autonomous Robotics</b>					
<b>Modul-Nr:</b> 310036 (WS), 310536 (SS)	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Jedes Semester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Praktikum/Projektarbeit			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Vorlesung Autonomous Robotics: Action, Perception, and Cognition</p>					
<p><b>Lernziele (learning outcomes):</b></p> <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> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Literaturrecherche und Dokumentation</li> <li>• Selbständiges Lernen und Arbeiten</li> <li>• Teamarbeit und Teamfähigkeit</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Programmieren</li> </ul> <p>Technologien:</p> <ul style="list-style-type: none"> <li>• Matlab</li> </ul>					
<p><b>Inhalt:</b></p> <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 Programmier-</p>					

## KAPITEL 5. NEUROINFORMATIK

umgebung MATLAB gelöst. Der theoretische Hintergrund der Lösungen ist der dynamische Systeme Ansatz der kognitiven Robotik.

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.

**Lehrformen:**

Praktikum/Projektarbeit

**Prüfungsformen:**

Übungsaufgaben

**Voraussetzungen für die Vergabe von Kreditpunkten:**

erfolgreiche Bearbeitung der Programmieraufgaben; Abgabe von schriftlichen Dokumentationen

**Stellenwert der Note für die Endnote:** 3 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Gregor Schöner**

Mitarbeiter des Instituts

**Sonstige Informationen:****Literatur:**

1. Valentino Braitenberg: Vehicles. Experiments in Synthetic Psychology, MIT Press, Cambridge, Mass 1984
2. Estela Bicho and Gregor Schöner: The dynamical approach to autonomous robotics demonstrated on a low-level vehicle platform. Robotics and Autonomous Systems 21:23-35 (1997)

<b>Deep Learning</b>					
<b>Modul-Nr:</b> 150332	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes WiSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  b) Übung			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<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> Folien, Tafelanschrieb					
<b>Prüfungsformen:</b> Abschlussprüfung; schriftlich, 120 Minuten					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. IT-Sicherheit / Informationstechnik, M.Sc. IT-Sicherheit / Netze und Systeme					
<b>Stellenwert der Note für die Endnote:</b> 5 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Jun. Prof. Dr. Asja Fischer					

# Kryptologie und Theoretische Informatik

<b>Kryptanalyse 1 (Einführung in die asymmetrische Kryptanalyse)</b>					
<b>Modul-Nr:</b> 150314	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Inhalte der Vorlesungen Einführung in die Kryptographie 1 und 2.</p>					
<p><b>Lernziele (learning outcomes):</b> Die Studierenden beherrschen die grundlegenden Algorithmen in der Kryptanalyse.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• IT-Sicherheit</li> <li>• Algorithmik</li> <li>• Mathematik</li> </ul>					
<p><b>Inhalt:</b> Die Vorlesung gibt einen Einblick in grundlegende Methoden der Kryptanalyse. Der Stoffplan umfasst die folgenden Themen:</p>					

## KAPITEL 6. KRYPTOLOGIE UND THEORETISCHE INFORMATIK

- Brute Force und Geburtstagsangriffe
- Time-Memory Tradeoffs
- Seitenkanalangriffe
- Gittertheorie und der LLL-Algorithmus
- Gitterbasierte Angriffe auf RSA
- Hidden Number Problem und Angriffe auf DSA
- Faktorisieren mit Faktorbasen
- Diskreter Logarithmus, Index-Calculus

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

Klausur

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Alexander May**

**Sonstige Informationen:**

Diese Veranstaltung wird mit 4+2 SWS in der ersten Hälfte des Semesters angeboten.

**Literatur:** Skript zur Vorlesung.

<b>Symmetrische Kryptanalyse</b>					
<b>Modul-Nr:</b> 150351	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes WiSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  b) Übungen			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Einführung in die Kryptographie 1					
<b>Lernziele (learning outcomes):</b> Die Studierenden haben ein vertieftes Verständnis für die Sicherheit symmetrischer Chiffren.					
<b>Inhalt:</b> Wir behandeln die wichtigsten Themen in der symmetrischen Kryptanalyse. Nach einer ausführlichen Vorstellung von linearer und differentieller Kryptanalyse werden weitere Angriffe auf symmetrische Primitive, insbesondere Block-Chiffren behandelt. Hierzu zählen insbesondere Integral (auch Square) Attacks, Impossible Differentials, Boomerang-Angriffe und Slide-Attacks. Neben den Angriffen selbst werden auch immer die daraus resultierenden Design-Kriterien beschrieben, um neue Algorithmen sicher gegen die Angriffe zu machen.					
<b>Lehrformen:</b> Vorlesungen und Übungen					
<b>Prüfungsformen:</b> Abschlussprüfung; mündlich, 30 Minuten					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene mündliche Prüfung					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. IT-Sicherheit / Informationstechnik, M.Sc. IT-Sicherheit / Netze und Systeme, B.Sc. Mathematik, M.Sc. Mathematik					
<b>Stellenwert der Note für die Endnote:</b> 5 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Prof. Dr. Gregor Leander					
<b>Sonstige Informationen:</b> <b>Literatur:</b> Knudsen, Lars, Robshaw, Matthew "The Block Cipher Companion", Springer, 2012					

<b>Geometrische Algorithmen</b>					
<b>Modul-Nr:</b> 150341	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Jedes zweite Semester, idR im WiSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung als kombinierter Folien- und Tafelvortrag, Übungen mit begleitendem Implementierungsprojekt			<b>Kontaktzeit:</b> 6 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppengröße:</b> 20
<b>Teilnahmevoraussetzungen:</b> Grundlegende Kenntnisse ueber Algorithmen und Datenstrukturen.					
<b>Empfohlene Vorkenntnisse:</b> 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 Quadtrees. 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, Übungen mit begleitendem Implementierungsprojekt					
<b>Prüfungsformen:</b>					



## KAPITEL 6. KRYPTOLOGIE UND THEORETISCHE INFORMATIK

Abschlussprüfung; Mündliche Prüfung
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene mündliche Prüfung, sowie als Studienleistung erfolgreiches Bearbeiten der Übungen und des Projektes mit Abgabe eines Projektberichtes
<b>Verwendung des Moduls (in anderen Studiengängen):</b> B.Sc. Mathematik, M.Sc. Mathematik
<b>Stellenwert der Note für die Endnote:</b> 6 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Maike Buchin</b>
<b>Sonstige Informationen:</b> <b>Literatur:</b> Die Vorlesung orientiert sich groesstenteils an dem Buch "Computational Geometry: Algorithms and Applications", von Mark de Berg, Otfried Cheong, Marc van Kreveld, und Mark Overmars (3te Auflage, 2008, Springer).

<b>Kryptographie</b>					
<b>Modul-Nr:</b> 150312 (Übung: 150313)	<b>Credits:</b> 8 CP	<b>Workload:</b> 240 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 6 SWS	<b>Selbststudium:</b> 150 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Keine					
<p><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.</p> <p><b>Vermittelte Kompetenzen:</b> Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• IT-Sicherheit</li> <li>• Algorithmik</li> <li>• Mathematik</li> </ul>					
<p><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 Komplexitätsannahmen in diesem Angreifermodell nachgewiesen.</p> <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> </ul>					

## KAPITEL 6. KRYPTOLOGIE UND THEORETISCHE INFORMATIK

- Konstruktion von Zufallszahlengeneratoren
- Diffie-Hellman Schlüsselaustausch
- Trapdoor Einwegpermutationen
- Public Key Verschlüsselung: RSA, ElGamal, Goldwasser-Micali, Rabin, Paillier
- Einwegsignaturen
- Signaturen aus kollisionsresistenten Hashfunktionen
- Random-Oracle Modell

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

Klausur

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene Modulabschlussklausur

**Stellenwert der Note für die Endnote:** 8 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Alexander May**

**Sonstige Informationen:**

Diese Veranstaltung ist im Vorlesungsverzeichnis der Mathematik als "Kryptographie I + II" aufgeführt.

**Literatur:** Katz, Lindell, "Introduction to Modern Cryptography", Chapman und Hall/CRC 2008

<b>Kryptographische Protokolle</b>					
<b>Modul-Nr:</b> 150343 (Übung: 150344)	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Inhalte des Moduls Kryptographie					
<p><b>Lernziele (learning outcomes):</b> Die Studierenden verstehen die erweiterten mathematischen Methoden und Verfahren, auf denen moderne kryptographische Protokolle beruhen. Die Teilnehmer sind zur Analyse und dem Design aktueller und zukünftiger kryptographischer Methoden befähigt.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Selbständiges Lernen und Arbeiten</li> <li>• Wissenschaftliches Arbeiten und Schreiben</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Algorithmik</li> <li>• IT-Sicherheit</li> <li>• Mathematik</li> </ul>					
<p><b>Inhalt:</b> Die Vorlesung beschäftigt sich mit erweiterten kryptographischen Protokollen und deren Anwendungen. Themenübersicht:</p> <ul style="list-style-type: none"> <li>• Identity-based Encryption</li> <li>• Digital Signatures</li> <li>• Secret sharing</li> <li>• Threshold Cryptography</li> <li>• Secure Multiparty Computation</li> </ul>					

## KAPITEL 6. KRYPTOLOGIE UND THEORETISCHE INFORMATIK

<b>Lehrformen:</b> Vorlesung (+Übung)
<b>Prüfungsformen:</b> Klausur
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Stellenwert der Note für die Endnote:</b> 5 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Prof. Dr. Eike Kiltz

<b>Systemsicherheit</b>					
<b>Modul-Nr:</b> 141340	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Erfahrung in systemnaher Programmierung sowie der Programmiersprache C sind hilfreich für das Verständnis der vermittelten Themen.</p>					
<p><b>Lernziele (learning outcomes):</b> Die Studierenden beherrschen theoretische und praktische Aspekte der Sicherheit von Softwaresystemen und sind zu einer kritischen Betrachtung der Systemsicherheit in der Lage. Insbesondere erwerben die Studierenden die Fähigkeit zum Modellieren konkreter Fragestellungen und Anforderungsanalysen aus vorhandenen Systeminformationen bzw. Systemgegebenheiten. Sie können eigenständig neue Probleme analysieren und neue Lösungsmöglichkeiten entwickeln. Die Studierenden sind in der Lage, neue Sicherheitsmodelle selbst zu erstellen und diese argumentativ zu verteidigen.</p>					
<p><b>Inhalt:</b> Im Rahmen der Vorlesung werden wichtige theoretische und praktische Aspekte aus dem Bereich der Systemsicherheit vorgestellt und diskutiert. Der Fokus liegt dabei auf verschiedenen Aspekten der Softwaresicherheit und verschiedene Angriffs- und Verteidigungstechniken werden vorgestellt. Die Studierenden sollen am Ende der Vorlesungsreihe in die Lage sein, die Sicherheit verschiedener Softwaresysteme zu analysieren, Schwachstellen im Design und der Implementierung aufzudecken sowie selbständig Sicherheitsmechanismen zu entwickeln. Darüber hinaus werden auch andere Aspekte aus dem Bereich der Systemsicherheit wie Privatheit und Anonymität betrachtet.</p>					
<p><b>Lehrformen:</b> Vorlesung (+Übung)</p>					
<p><b>Prüfungsformen:</b> Schriftliche Modulabschlussprüfung</p>					
<p><b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur</p>					
<p><b>Stellenwert der Note für die Endnote:</b> 5 / 112</p>					
<p><b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Thorsten Holz</b></p>					

<b>Model Checking</b>					
<b>Modul-Nr:</b> 150324	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes WiSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung mit begleitenden Übungen			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine  <b>Empfohlene Vorkenntnisse:</b></p> <ul style="list-style-type: none"> <li>• Grundlagenvorlesungen Mathematik,</li> <li>• Einführung in die Theoretische Informatik (ggf. kann das nötige Wissen auch nachgeholt werden),</li> <li>• Hilfreich: Logik in der Informatik, Datenstrukturen und elementare Programmierkenntnisse</li> </ul>					
<p><b>Lernziele (learning outcomes):</b>            Die Studierenden lernen wie sich verteilte Systeme durch Transitionssysteme modellieren und Eigenschaften in logischen Spezifikations-sprachen wie LTL und CTL spezifizieren lassen. Sie sollen elementare Algorithmen zur Überprüfung von Eigenschaften in Transitionssystemen kennenlernen. Sie sollen ein Verständnis für die Möglichkeiten und Grenzen des Model Checking entwickeln, und in die Lage versetzt werden, sich eigenständig mit fortgeschrittenen Methoden des Model Checkings auseinanderzusetzen.</p>					
<p><b>Inhalt:</b>            Wie kann die Korrektheit von Software und Hardware formal überprüft werden? Im Model Checking werden Software- und Hardware-Module durch Transitionssysteme formalisiert; gewünschte Eigenschaften mit Hilfe logischer Formalismen formal beschrieben; und mit Hilfe von Algorithmen automatisiert überprüft, ob ein Transitionssystem eine formal spezifizierte Eigenschaft besitzt. In dieser Veranstaltung werden die theoretischen Grundlagen des Model Checkings vermittelt, mit einem Fokus auf logik-basierten Spezifikations-sprachen. Die Spezifikations-sprachen LTL und CTL werden eingeführt, ihre Ausdrucksstärke untersucht, und die wichtigsten algorithmischen Ansätze für das Model Checking vorgestellt.</p>					
<p><b>Lehrformen:</b>            Vorlesung mit begleitenden Übungen</p>					
<p><b>Prüfungsformen:</b>            Abschlussprüfung; mündliche Prüfung (20-30min) oder schriftliche Klausur (120min) in Abhängigkeit der Teilnehmerzahl</p>					
<p><b>Voraussetzungen für die Vergabe von Kreditpunkten:</b>            bestandene mündliche oder schriftliche Prüfung</p>					

<b>Verwendung des Moduls (in anderen Studiengängen):</b> Studiengang Mathematik, Informatik, IT-Sicherheit
<b>Stellenwert der Note für die Endnote:</b> 5 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Jun.-Prof. Thomas Zeume</b>
<b>Sonstige Informationen:</b> Veranstaltung findet nicht im WS 22/23 statt und ab SS 23 jeweils im SS  <b>Literatur:</b> Einstiegsliteratur für diese Veranstaltung sind die Bücher: <ul style="list-style-type: none"><li>• Baier, Christel, and Joost-Pieter Katoen. Principles of model checking. MIT press, 2008.</li><li>• Clarke Jr, E. M., Grumberg, O., Kroening, D., Peled, D., und Veith, H. Model checking. MIT press.2018.</li></ul>



# Wirtschaftsinformatik und Management

<b>Industrielles Kundenmanagement</b>					
<b>Modul-Nr:</b> 139010	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Industrielles Kunden- Management			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> 20 Studierende
<b>Teilnahmevoraussetzungen:</b> Keine					
<p><b>Lernziele (learning outcomes):</b>            Zielsetzung: Die Studierenden erwerben fachspezifische Kompetenzen in Form von Kenntnissen zu Grundlagen von an den Maschinenbau angrenzenden, relevanten Ingenieurwissenschaften, wie dem Wirtschaftsingenieurwesen und darüber hinaus Ansätze der Wirtschaftspsychologie und wirtschaftswissenschaftliche Aspekte. Die Studierenden sollen zentrale Methoden des Industriellen Kunden-Managements in den relevanten Management-Komponenten aktiv anwenden können und theoriegeleitet wissenschaftliche (Fall-)Studien bearbeiten, kritisch darstellen und präsentieren. Als generische Kompetenzen werden die Studierenden zu vernetztem Denken angeregt und in die Lage versetzt etablierte Methoden und Verfahren auszuwählen und anzuwenden und daraus kritisches Managementhandeln abzuleiten. Dies beinhaltet spezifische Kenntnisse und instrumentale Kompetenzen des industriellen Kunden-Managements aus prozessualer, organisatorischer und strategischer Sicht. Die Studierenden erlangen vertiefte, interdisziplinäre instrumentale Kompetenzen und können diese situativ anwenden. Dabei lernen die Studierenden die Grundlagen, Methoden und Verfahren der wirtschaftsingenieur- und verhaltenswissenschaftlichen Ansätze in Praxis und Forschung im Bereich des IKM kennen. Dies beinhaltet die Analyse, Bewertung und Gestaltung der Aufbau- und Ablauforganisation und der Geschäftsmodelle unterschiedlicher Organisationen sowie wesentliche Elemente der kundenorientierten Strategieentwicklung im Kontext einer digitalen Transformation der beruflichen und privaten Lebenswelt.</p>					
<p><b>Inhalt:</b>            In den Veranstaltungen zum Industriellen Kunden-Managements werden Theorien, Modelle und Ansätze dargestellt und anhand von Fallstudien und Business-Episoden vertieft und reflektiert. Die Veranstaltung orientiert sich an der zyklischen Bearbeitung von 12 wesentlichen</p>					

## KAPITEL 7. WIRTSCHAFTSINFORMATIK UND MANAGEMENT

Management-Kompetenzfeldern zum ganzheitlichen Kunden-Management.

12 Episoden bzw. Management-Kompetenzfelder:

1. Innovationsmanagement
2. Technologiemanagement
3. FundE Management
4. Prozessmanagement
5. Qualitätsmanagement
6. Produktmanagement
7. Marketingmanagement
8. Vertriebsmanagement
9. Servicemanagement
10. Projektmanagement
11. Change Management
12. Kundenmanagement

Arbeitsaufwände: - Präsenzzeit: 45 h Präsenzstudium - Vor und Nachbereitung (einschl. Prüfung): 105 h Eigenstudium

**Lehrformen:**

Unterricht in Seminarform, Gruppenarbeiten, Bearbeitung von Fallbeispielen

**Prüfungsformen:**

Gruppenarbeit mit Präsentation zur Erlangung von Bonuspunkten (freiwillig); Mündliche Abschlussprüfung in Kleingruppen von 3-4 Personen und einer Dauer von 20 Minuten pro Prüfling

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Teilnahme und bestehen der mündlichen Prüfung

**Verwendung des Moduls (in anderen Studiengängen):** Studiengang Maschinenbau Master; Optionalbereich

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. phil. Joachim Zülch**  
Dr.-Ing. Matthias Bartels

**Sonstige Informationen:**

Das Modul eignet sich für interessierte Studierende der Master-Phase im Optionalbereich und

## KAPITEL 7. WIRTSCHAFTSINFORMATIK UND MANAGEMENT

erfordert keine speziellen Vorkenntnisse. Weitere Informationen sowie die Unterlagen zu Vorlesungen und Übungen werden über Moodle zur Verfügung gestellt.

<b>Rationales Entscheiden</b>					
<b>Modul-Nr:</b> 074210 (Übung: 074211)	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Letztmalig im WS 20/21!	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 3 SWS	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Die Kenntnisse linearer Optimierungsmodelle und statistischer Grundlagen wird vorausgesetzt. Eine Auffrischung der Kenntnisse erfolgt im Rahmen eines Kolloquiums (1 SWS).</p> <p><b>Empfohlene Vorkenntnisse:</b> Betriebswirtschaftliche Grundkenntnisse sind hilfreich.</p>					
<p><b>Lernziele (learning outcomes):</b> Absolventen dieser Veranstaltung sind in der Lage, komplexe Entscheidungssituationen zu strukturieren und selbständig entscheidungstheoretisch fundierte Vorschläge für gute Lösungen zu entwickeln. Dieses allgemeine methodisch orientierte betriebswirtschaftliche Modul eignet sich besonders als theoretische Vorbereitung für die analytische strukturierte Behandlung komplexer Probleme.</p> <p><b>Vermittelte Kompetenzen:</b> Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Selbständiges Lernen und Arbeiten</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Optimierungsmethoden</li> <li>• Wirtschaftliches Handeln</li> </ul>					
<p><b>Inhalt:</b> Es werden grundlegende entscheidungs- und informationstheoretische Konzepte vorgestellt und anhand wirtschafts-wissenschaftlicher Beispiele verdeutlicht. Besonders berücksichtigt werden Entscheidungen unter Unsicherheit, also Risiko und Ungewissheit, im Unterschied zu solchen unter Sicherheit. Weiter werden Entscheidungen mit mehreren Zielen oder mit mehreren Entscheidungsträgern ausführlich behandelt. Dynamische Entscheidungen und weiterführende Ansätze werden ebenfalls vorgestellt.</p>					
<p><b>Lehrformen:</b> Vorlesung (+Übung)</p>					
<p><b>Prüfungsformen:</b> Klausur 90 Minuten</p>					

<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur
<b>Stellenwert der Note für die Endnote:</b> 5 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Brigitte Werners</b>
<b>Sonstige Informationen:</b> Letztmalig im WS 20/21  <b>Literatur:</b> Veranstaltungsunterlagen und Literaturhinweise werden über Blackboard bereitgestellt

<b>Managing Digital Platform Ecosystems</b>					
<b>Modul-Nr:</b> MISSING	<b>Credits:</b> 7 CP	<b>Workload:</b> 210 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Lernziele (learning outcomes):</b> After completion of this module, students will understand the basic concepts of platforms as two-sided markets, model platform ecosystems, understand the design and governace of digital platforms from the perspective of the platform owner and analyze existing platform ecosystems.					
<b>Inhalt:</b> In this module, students will learn about digital platform ecosystems. Digital platforms are at the heart of business model change in digital transformation. Companies like Apple, Facebook, Uber and Google, but also Microsoft, SAP and Salesforce develop ecosystems, where partners develop a plethora of applications to match customer needs on their digital platform. Digital platform ecosystems will be analyzed from both thenological design as well as the market perspective. The course addressess the managerial challenges in building and scaling platform-based business models and governing the ecosystem surrounding the digital platform. In addition to participating in the lecture, students will conduct a group case work assignment on digital platform ecosystems.					
<b>Lehrformen:</b> Vorlesung					
<b>Prüfungsformen:</b> Semesterbegleitend; Module examination, consisting of an oral exam or a graded written exam (60 minutes) (counts for 60 percent of the grading), both of them in combination with the preparation and presentation of a case (counts for 40 percent of the grading) (mode will be announced in time).					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Elective Module in the TU Dortmund M.Sc. programmes Wirtschaftswissenschaften, WiWi für ein Lehramt am Berufskolleg (Modellversuch), Wirtschaftsmathematik, Wirtschaftsingenieurwesen, Logistik, Mathematik und Statistik.					
<b>Stellenwert der Note für die Endnote:</b> 7 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>					



# Bioinformatik

<b>Bioinformatics for Proteomics I</b>					
<b>Modul-Nr:</b> 201911	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung  b) Übung			<b>Kontaktzeit:</b> V: 2 SWS; Ü: 1	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> 3 bis 20 Studie- rende
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Recommended prior knowledge: English and basic programming skills.					
<p><b>Lernziele (learning outcomes):</b> Discipline-specific competences: After the successful completion of this module:</p> <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 use proteomics-specific software and the workflow engine KNIME,</li> <li>• they are able to design and program own solution strategies</li> <li>• and they are able to apply the discussed software tools and methods to real data and problems.</li> </ul> <p>Interdisciplinary/generic competences:</p> <ul style="list-style-type: none"> <li>• <b>instrumental competences:</b></li> </ul>					



- 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:**

- Basics of protein biochemistry
- Properties of amino acids
- Basics of mass spectrometry
- Raw data processing
- Protein databases
- Tryptic and in silico digest of proteins
- 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
- Preprocessing of quantitative data
- Quality control
- Software tools used in bioinformatics for proteomics (Tutorial)
- Practical (programming) tasks (Tutorial)
- Workflow engine KNIME (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 test.

## KAPITEL 8. BIOINFORMATIK

<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Passed oral or written examination.
<b>Stellenwert der Note für die Endnote:</b> 5 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  PD Dr. Martin Eisenacher
<b>Sonstige Informationen:</b> Weitere Informationen unter <a href="http://www.bioinf.rub.de/msc/">http://www.bioinf.rub.de/msc/</a>

<b>Bioinformatics for Proteomics II</b>					
<b>Modul-Nr:</b> 201911	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Sommersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Vorlesung  b) Übung			<b>Kontaktzeit:</b> V: 2 SWS; Ü: 1	<b>Selbststudium:</b> 105 h	<b>Gruppen- größe:</b> 3 bis 20 Studie- rende
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Recommended prior knowledge: English, basic programming skills and lecture/tutorials “Bioinformatics for Proteomics I” in the winter term (recommended, not required).</p>					
<p><b>Lernziele (learning outcomes):</b> Discipline-specific competences: After the successful completion of this module</p> <ul style="list-style-type: none"> <li>• the students have become familiar with the most important knowledge from the lecture “Bioinformatics for Proteomics I” as a brief recapitulation,</li> <li>• they have become familiar with the principles of advanced methods used in bioinformatics for proteomics,</li> <li>• 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,</li> <li>• they understand the underlying algorithmic and statistical concepts of these methods,</li> <li>• they are able to use proteomics-specific software and the workflow engine KNIME,</li> <li>• they are able to design and program own solution strategies</li> <li>• and they are able to apply the discussed software tools and methods to real data and problems.</li> </ul> <p>Interdisciplinary/generic competences:</p> <ul style="list-style-type: none"> <li>• <b>instrumental competences:</b> <ul style="list-style-type: none"> <li>– Intensive usage of the learning platform Moodle</li> </ul> </li> <li>• <b>systemic competences:</b> <ul style="list-style-type: none"> <li>– Independent learning and working</li> <li>– Teamwork and ability to work in a team</li> </ul> </li> </ul>					

• **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”
- Computational comparison of protein lists
- Statistics for the comparison of experimental groups
- Machine learning-based biomarker discovery (supervised and unsupervised methods)
- Enrichment analysis
- Network analysis
- Single / multiple / parallel reaction monitoring (SRM / MRM / PRM)
- Data independent acquisition (DIA)
- Algorithms for de novo sequencing of peptides
- Open searches
- Dark matter of proteomics
- Proteoforms
- Metaproteomics and proteogenomics
- Software tools used in bioinformatics for proteomics (Tutorial)
- Practical (programming) tasks (Tutorial)
- Workflow engine KNIME (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 test

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Passed oral or written examination.

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**PD Dr. Martin Eisenacher**

**Sonstige Informationen:**

Weitere Informationen unter <http://www.bioinf.rub.de/msc/>

<b>Bioimage Informatics</b>					
<b>Modul-Nr:</b> 190800	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vorlesung (+Übung)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine <b>Empfohlene Vorkenntnisse:</b> Grundlegende Programmierkenntnisse					
<p><b>Lernziele (learning outcomes):</b> Die Studierenden sollen elementare und aktuelle Techniken der Bildverarbeitung kennenlernen, die zur Analyse mikroskopischer Bilddaten verwendet werden. Sie sollen die dahinterliegenden algorithmischen, mathematischen und statistischen Ideen verstehen, und lernen in der Praxis, wie diese Methoden auf reale Daten und Fragestellungen angewendet werden.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Algorithmen</li> <li>• Programmieren</li> </ul>					
<p><b>Inhalt:</b></p> <p>Die Analyse von mikroskopischen Bilddaten mit Methoden der Bildverarbeitung ist in den vergangenen Jahren ein wichtiges Thema in vielen Anwendungen in den Lebenswissenschaften geworden. In der Vorlesung werden grundlegende Konzepte der Verarbeitung mikroskopischer Bilddaten und deren Anwendungen behandelt. Die Vorlesung gliedert sich wie folgt:</p> <ul style="list-style-type: none"> <li>• Grundlagen der Mikroskopie: Durchlicht- und Konfokalmikroskopie; Fluoreszenzmikroskopie; Mikroskopie in der Histopathologie; Mikroskopie in der Zellbiologie; Mikroskopie in der Neurobiologie</li> <li>• Morphologische Bildanalyse</li> <li>• Texturelle Bildanalyse</li> <li>• Kolokalisations-Verfahren</li> <li>• Algorithmen zur Bildregistrierung</li> <li>• Algorithmen zum Verfolgen von Bewegungsmustern von Zellen (“Cell Tracking”)</li> <li>• Verfahren zur Rekonstruktion von Neuronen (“Neuron Tracing”)</li> </ul>					

## KAPITEL 8. BIOINFORMATIK

- Deep Learning Methoden zur mikroskopischen Bilddatenanalyse
- Methoden der Analyse von markerfreien Mikroskopie-Daten
- Methoden zur überwachten und unüberwachten Segmentierung von mikroskopischen Bildern

**Lehrformen:**

Vorlesung (+Übung)

**Prüfungsformen:**

mündliche Prüfung

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Bestandene mündliche Prüfung

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**PD Dr. Axel Mosig**

**Sonstige Informationen:**

Weitere Informationen unter <http://www.bioinf.rub.de/msc/>

<b>Master-Praktikum Big Data in der Bioinformatik</b>					
<b>Modul-Nr:</b> 202621	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> immer	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Praktikum			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> 1-3 Studierende
<p><b>Teilnahmevoraussetzungen:</b> praktische Programmiererfahrung in einer oder mehreren Sprachen (z. B. Java, R, Python, Matlab)</p> <p><b>Empfohlene Vorkenntnisse:</b> Vertiefungsmodul(e) aus dem Themengebiet der Bioinformatik</p>					
<p><b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Moduls</p> <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> <p>Vermittelte Kompetenzen: Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Programmieren von Lösungen für bioinformatische Anwendungen</li> <li>• Umgang mit Standard-Werkzeugen / -Formaten und Programmier-Bibliotheken der Bioinformatik</li> <li>• Umgang mit größeren Datenmengen</li> </ul> <p>Generische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• <b>instrumentale Kompetenzen</b> <ul style="list-style-type: none"> <li>– Projekt- und Zeitmanagement mit digitalen Tools</li> <li>– Wissenschaftlich gegliederte Dokumentation</li> </ul> </li> <li>• <b>systemische Kompetenzen</b></li> </ul>					



- Teamarbeit und Teamfähigkeit
- Selbständiges Lernen und Arbeiten

• **kommunikative Kompetenzen**

- Präsentation wissenschaftlicher Ergebnisse
- Vermittlung der in der Bioinformatik angewandten englischen Fachbegriffe
- Rhetorik und sprachliche Kompetenz (deutsch / englisch)

**Inhalt:**

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.

**Lehrformen:**

Einführung als seminaristischer Unterricht, Bearbeitung der praktischen Aufgabe selbständig oder als Gruppenarbeit

**Prüfungsformen:**

Semesterbegleitend; Protokoll max. 10 Seiten (wenn Abgabe bis zwei Wochen nach Praktikumsende, dann Korrekturrunde vor Benotung), Dokumentation der praktischen Aufgabe

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Protokoll (siehe Prüfungsformen), Bearbeitung / Dokumentation der praktischen Aufgabe, Teilnahme an den Präsenztreffen, Abschlusspräsentation (15 min. + 5 min. Diskussion)

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Axel Mosig**

Prof. Dr. Axel Mosig, Prof. Dr. Sven Rahmann (Universität Duisburg/Essen und TU Dortmund)

**Sonstige Informationen:**

Weitere Informationen unter <http://www.bioinf.rub.de/msc/>

<b>Master-Praktikum Computational Proteomics</b>					
<b>Modul-Nr:</b> 202621	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> jedes SoSe	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Praktikum (202621)			<b>Kontaktzeit:</b> 4 SWS	<b>Selbststudium:</b> 90 h	<b>Gruppen- größe:</b> 1-3 Studierende
<b>Teilnahmevoraussetzungen:</b> Practical programming experience in one or more languages (e. g. Java, R, Python, Matlab)					
<b>Empfohlene Vorkenntnisse:</b> Modules from the focal point Bioinformatics (e.g. lecture “Bioinformatics for Proteomics I”)					
<b>Lernziele (learning outcomes):</b> After successful completion of the module students:					
<ul style="list-style-type: none"> <li>• can take on practical challenges in the development of bioinformatics applications,</li> <li>• the students have got to know programming libraries from the field of bioinformatics</li> <li>• have practiced the use of workflow systems</li> <li>• have learned the mechanisms of code and program documentation</li> <li>• have practiced procedures used to provide own libraries (e.g. R packages, .jar files)</li> </ul>					
Discipline-specific competences:					
<ul style="list-style-type: none"> <li>• Programming of solutions for bioinformatics applications</li> <li>• Handling of standard tools / formats and programming libraries of bioinformatics</li> <li>• Dealing with large amounts of data</li> </ul>					
Generic competences:					
<ul style="list-style-type: none"> <li>• <b>Instrumental competences</b> <ul style="list-style-type: none"> <li>– Project and time management with digital tools</li> <li>– Scientifically structured documentation</li> </ul> </li> <li>• <b>Systemic competences</b> <ul style="list-style-type: none"> <li>– Teamwork and ability to work in a team</li> <li>– Independent learning and working</li> </ul> </li> <li>• <b>Communicative skills</b></li> </ul>					

## KAPITEL 8. BIOINFORMATIK

- Presentation of scientific results
- Using English technical terms used in Bioinformatics
- Rhetoric and linguistic competence (German / English)

**Inhalt:**

Bioinformatics naturally applies computer science methods to the data of a life science application area, i. e. it represents applied digitization per se. The internship teaches the basics of programming / development with relations life science applications utilizing data of high-throughput mass spectrometry / Proteomics. After a short introduction (presence meeting) in programming and development environments (e.g. Java, R, C++, eclipse, RStudio, Python, Matlab, KNIME), practical programming tasks will be enrolled and edited during the semester, if necessary with further face-to-face meetings to discuss progress.

**Lehrformen:**

Introductory weeks, independent working on the practical task alone or as group work

**Prüfungsformen:**

Semesterbegleitend; Written protocol with the practical's topics and the work's results (max. 10 pages), documentation of the practical task, final slide presentation (15 min. + 5 min. discussion)

**Voraussetzungen für die Vergabe von Kreditpunkten:**

Written protocol (see "Prüfungsformen"), processing / documentation of the practical task, participation in the face-to-face meetings, final presentation

**Stellenwert der Note für die Endnote:** 5 / 112

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**PD Dr. Martin Eisenacher**

# Masterseminare

<b>Seminar Ingenieurinformatik</b>					
<b>Modul-Nr:</b> 125017	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> 4.-6.. Sem.	<b>Turnus:</b> Winter- und Sommersemes- ter	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Seminar Ingenieurinformatik			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Mindestens ein Vertiefungsmodul der Ingenieurinformatik</p>					
<p><b>Lernziele (learning outcomes):</b> Lernziel ist die selbständige Auseinandersetzung mit einem Thema aus dem Bereich der Ingenieurinformatik. Es soll die Fähigkeit der kritischen Auseinandersetzung mit einem Thema im Rahmen einer Fachdiskussion gefördert werden. Die Studierenden vertiefen sich in ein Gebiet, das ihnen zu Beginn des Seminars zugewiesen wird. Der Inhalt muß verständlich und fachlich korrekt wiedergegeben werden. Anschließende Diskussionen aller Seminarteilnehmer sollen dazu führen, dass sich die Studierenden kritisch mit dem fachlichen Inhalt des Vortrages auseinandersetzen.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> <li>• kennen Studierende die methodischen Ansätze und Techniken, um sich in eine Thematik gründlich einzuarbeiten. Dabei sollen die gewonnenen Erkenntnisse in verständlicher Form weitergegeben werden;</li> <li>• verstehen Studierende durch Bearbeitung einer Seminararbeit, wie die erarbeiteten Fakten, Informationen und Zusammenhänge in Rahmen einer Präsentation klar vermittelt werden können.</li> </ul> <p><b>Vermittelte Kompetenzen:</b> Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> </ul>					

## KAPITEL 9. MASTERSEMINARE

- Kritikfähigkeit
- Literaturrecherche und Dokumentation
- Projekt- und Zeitmanagement
- Präsentation wissenschaftlicher Ergebnisse
- Rhetorik und sprachliche Kompetenz
- Selbständiges Lernen und Arbeiten

### **Inhalt:**

Die im Rahmen eines Semesters angebotenen Seminarthemen werden zu Beginn des Semesters bekannt gegeben und decken Themen auf dem Gebiet der Ingenieurinformatik ab. Es wird darauf geachtet, dass die Themen einen engen Bezug zu aktuellen Problemstellungen, dem Stand der Technik und neuen Forschungserkenntnissen haben. Folgende Themenschwerpunkte werden behandelt

- Product Lifecycle Management
- Smart Product Engineering
- Product Development Methods
- Echtzeit Rendering
- Visual Analytics
- Simulationstechniken

### **Lehrformen:**

seminaristischer Unterricht; Erstellen einer Seminararbeit

### **Prüfungsformen:**

Semesterbegleitend; Präsentation der eigenen Seminararbeit

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

- Erfolgreiche Bearbeitung einer Seminararbeit
- aktive Beteiligung mit Diskussionen an allen Präsentationen

**Stellenwert der Note für die Endnote: 3 / 112**

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

**Prof. Dr. Markus König**  
Dr. K. Lehner

<b>Seminar Kryptologie und Theoretische Informatik</b>					
<b>Modul-Nr:</b> 150537: Se- minar zur Kryptographie	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> BA: 4.-6.; MA: 2.-3.. Sem.	<b>Turnus:</b> Unregelmäßig	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vertiefungsseminare			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Kryptographie, Kryptanalyse, Algorithmik					
<b>Lernziele (learning outcomes):</b> Es werden aktuelle Forschungsarbeiten der Kryptologie und Algorithmik vorgestellt					
<b>Vermittelte Kompetenzen:</b> Kernkompetenzen:					
<ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Kritikfähigkeit</li> <li>• Literaturrecherche und Dokumentation</li> <li>• Projekt- und Zeitmanagement</li> <li>• Präsentation wissenschaftlicher Ergebnisse</li> <li>• Rhetorik und sprachliche Kompetenz</li> <li>• Selbständiges Lernen und Arbeiten</li> </ul>					
<b>Inhalt:</b> Inhalt entsprechend des gewählten Seminars.					
<b>Lehrformen:</b> Vertiefungsseminare					
<b>Prüfungsformen:</b> Semesterbegleitend; Seminar-Arbeit					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Erfolgreicher Seminarvortrag					
<b>Stellenwert der Note für die Endnote:</b> 3 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Prof. Dr. Alexander May					

## KAPITEL 9. MASTERSEMINARE

Prof. Dr. Alexander May, Prof. Dr. Hans Ulrich Simon, Prof. Dr. Eike Kiltz

**Sonstige Informationen:**

Angebot siehe VVZ

<b>Seminar Computerlinguistik</b>					
<b>Modul-Nr:</b> 050025: Tools für den linguistischen Alltag: Automatisierung, Reproduzierbarkeit und Kollaboration; 050042: Dialogsysteme; 050042: Koreferenzauflösung	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> BA: 4.-6.; MA: 2.-3.. Sem.	<b>Turnus:</b> Sommersemester und Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vertiefungsseminare			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppengröße:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Linguistische Grundlagen, Linguistische Schwerpunkte und Linguistische Methoden					
<b>Lernziele (learning outcomes):</b>					
<ol style="list-style-type: none"> <li>1. Fortgeschrittene Kenntnisse in unterschiedlichen Bereichen der Computerlinguistik und Sprachtechnologie</li> <li>2. Fähigkeit, die für computerlinguistische Themen und Aufgabenstellungen jeweils geeigneten Analysemethoden sowie die entsprechenden Algorithmen und Implementationsverfahren auswählen, bewerten, und anwenden zu können</li> <li>3. generelle Kompetenz in eigenständiger computerlinguistischer Analyse, Modellierung und Implementierung sprachlicher Strukturen und Prozesse</li> </ol>					
<b>Vermittelte Kompetenzen:</b>					
Kernkompetenzen:					
<ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Literaturrecherche und Dokumentation</li> <li>• Projekt- und Zeitmanagement</li> <li>• Präsentation wissenschaftlicher Ergebnisse</li> <li>• Selbständiges Lernen und Arbeiten</li> </ul>					



<p><b>Inhalt:</b></p> <p>Gegenstand des Moduls sind unterschiedliche Themen aus den Bereichen Computerlinguistik und Sprachtechnologie, sprich deren spezifische Untersuchungsgegenstände und Methoden.</p> <p><b>Seminare:</b> Typische Themen einschlägiger und in unterschiedlichem Turnus angebotener Seminare sind u. a.</p> <ul style="list-style-type: none"> <li>• Korpuslinguistik</li> <li>• Data mining, Explorative Datenanalyse (mit R)</li> <li>• Klassifikationsalgorithmen</li> <li>• Natürlichsprachliche Systeme, Automatische Sprachverarbeitung, Automatische Lexikonakquise</li> <li>• Automatische Textzusammenfassung, Textgenerierung</li> <li>• Semantische Verarbeitung in der Computerlinguistik</li> <li>• Morphologische und syntaktische Analyse in der Computerlinguistik</li> </ul>
<p><b>Lehrformen:</b></p> <p>Vertiefungsseminare</p>
<p><b>Prüfungsformen:</b></p> <p>Semesterbegleitend; Seminar-Arbeit</p>
<p><b>Voraussetzungen für die Vergabe von Kreditpunkten:</b></p> <p>Bestandene Seminar-Arbeit</p>
<p><b>Stellenwert der Note für die Endnote:</b> 3 / 112</p>
<p><b>Modulbeauftragte/r und hauptamtlich Lehrende:</b></p> <p><b>Johanna Marie Poppek, M. A.</b> Roussel, Simonjetz, Klabunde</p>
<p><b>Sonstige Informationen:</b></p> <p>Angebot siehe VVZ</p> <p>Die Teilnahme ist nur nach vorheriger Anmeldung in CampusOffice möglich. Diese Anmeldung muss im WS bis zum 01.10. und im SS bis zum 01.04. erfolgen. Zusätzlich ist die Anmeldung zur Prüfung über FlexNow innerhalb der gültigen Anmeldephase notwendig.</p> <p><b>Literatur:</b> Wird im Kurs bekanntgegeben.</p>

<b>Seminar Operations Research (Management Science)</b>					
<b>Modul-Nr:</b> WS: 074230, SS: 074232	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 2.-3.. Sem.	<b>Turnus:</b> Letztmalig im SS 21!	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vertiefungsseminare			<b>Kontaktzeit:</b> WS: 4, SS: 6	<b>Selbststudium:</b> 150 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Fundierte Kenntnisse der Anwendung quantitativer Methoden aus dem Bachelorstudium oder der Mastermodule "Rationales Entscheiden" und "Management Science" werden vorausgesetzt.</p> <p><b>Empfohlene Vorkenntnisse:</b> Betriebswirtschaftliche und entscheidungstheoretische Kenntnisse sowie Programmierkenntnisse sind hilfreich.</p>					
<p><b>Lernziele (learning outcomes):</b> Studierende erwerben vertiefte Kenntnisse quantitativer Methoden und Anwendungen zur optimalen Lösung vielfältiger wirtschaftlicher Fragestellungen, die teils von Praxispartner stammen. Gleichzeitig wird die Fähigkeit zur selbständigen Auseinandersetzung mit komplexen Problemen auf wissenschaftlichem Niveau und unter Berücksichtigung des aktuellen Stands der Forschung erreicht.</p> <p><b>Vermittelte Kompetenzen:</b></p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Analytisches und logisches Denken</li> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Selbständiges Lernen und Arbeiten</li> <li>• Kritikfähigkeit</li> <li>• Projekt- und Zeitmanagement</li> <li>• Rhetorik und sprachliche Kompetenz</li> <li>• Literaturrecherche und Dokumentation</li> <li>• Präsentation wissenschaftlicher Ergebnisse</li> <li>• Wissenschaftliches Arbeiten und Schreiben</li> </ul> <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> <li>• Optimierungsmethoden</li> <li>• Simulationsmethoden</li> <li>• Wirtschaftliches Handeln</li> </ul>					

## KAPITEL 9. MASTERSEMINARE

<p><b>Inhalt:</b></p> <p>Jeweils ausgewählte Gebiete des Management Science werden vertiefend behandelt. Dazu werden Themenstellungen durch die Teilnehmer selbständig bearbeitet und in Form einer Hausarbeit und einer Präsentation zur Diskussion gestellt. Damit erfolgt auch eine Vorbereitung auf die Masterarbeit, insbesondere im Bereich Operations and Service Management.</p> <p>Die Modulnote setzt sich aus der Note der Hausarbeit (70 %) und der für Präsentation und Diskussion im Plenum (30 %) zusammen.</p> <p><b>Seminare:</b> Angeboten über die Fakultät für Wirtschaftswissenschaft Vorlesungsnr.: 074231 (Sommersemester 2013) Vorlesungsnr.: 074232 (Wintersemester 2013/14) Modul Seminar Management Science geplantes Angebot jeweils im Sommer- und Wintersemester</p>
<p><b>Lehrformen:</b></p> <p>Vertiefungsseminare</p>
<p><b>Prüfungsformen:</b></p> <p>Semesterbegleitend; Seminar-Arbeit</p>
<p><b>Voraussetzungen für die Vergabe von Kreditpunkten:</b></p> <p>Bestandene Seminar-Arbeit</p>
<p><b>Stellenwert der Note für die Endnote:</b> 5 / 112</p>
<p><b>Modulbeauftragte/r und hauptamtlich Lehrende:</b></p> <p><b>Prof. Dr. Brigitte Werners</b></p>
<p><b>Sonstige Informationen:</b></p> <p>Letztmalig im SS 21!</p> <p><b>Literatur:</b> Veranstaltungsunterlagen und Literaturhinweise aus wissenschaftlichen Zeitschriften werden zu Beginn der Veranstaltung angekündigt bzw. bereitgestellt.</p>

<b>Seminar Bioinformatik</b>					
<b>Modul-Nr:</b> 190559	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> 2.-3.. Sem.	<b>Turnus:</b> Jedes Semester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Seminar Bioinformatik			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Mindestens ein Vertiefungsmodul auf dem Themengebiet der Bioinformatik</p>					
<p><b>Lernziele (learning outcomes):</b> Lernziel ist die selbständige Auseinandersetzung mit einem Thema aus dem Bereich der Bioinformatik. Es soll die Fähigkeit der kritischen Auseinandersetzung mit einem Thema im Rahmen einer Fachdiskussion gefördert werden.</p> <p><b>Vermittelte Kompetenzen:</b> Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Fächerübergreifendes Denken und Arbeiten</li> <li>• Kritikfähigkeit</li> <li>• Literaturrecherche und Dokumentation</li> <li>• Projekt- und Zeitmanagement</li> <li>• Präsentation wissenschaftlicher Ergebnisse</li> <li>• Rhetorik und sprachliche Kompetenz</li> <li>• Selbständiges Lernen und Arbeiten</li> </ul>					
<p><b>Inhalt:</b> Die Themen werden zu Semesterbeginn bekanntgegeben und werden jeweils aus aktuellen Themen der Bioinformatik zusammengestellt, insbesondere aus den Themengebieten</p> <ul style="list-style-type: none"> <li>• Bioinformatik der Proteomik</li> <li>• Bioinformatik der Sequenzanalyse</li> <li>• Analyse mikroskopischer Bilddaten</li> <li>• Analyse von Omics Daten</li> <li>• Bioinformatik und molekulare Evolution</li> </ul>					
<p><b>Lehrformen:</b> Seminar Bioinformatik</p>					

## KAPITEL 9. MASTERSEMINARE

<b>Prüfungsformen:</b> Semesterbegleitend; Seminararbeit
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Seminar-Arbeit
<b>Stellenwert der Note für die Endnote: 3 / 112</b>
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>PD Dr. Axel Mosig</b> PD Dr. Martin Eisenacher; Prof. Dr. Axel Mosig; Prof. Dr. Sven Rahmann

<b>Seminar Algorithmen</b>					
<b>Modul-Nr:</b> 125017	<b>Credits:</b> 5 CP	<b>Workload:</b> 150 h	<b>Semester:</b> 2-4.. Sem.	<b>Turnus:</b> unregelmäßig	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Seminar			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> 12
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> <li>• Können Studierende Literatur zu einem algorithmischem Thema sich selber erarbeiten</li> <li>• Können Studierende ein algorithmisches Thema mündlich vortragen</li> <li>• Haben Studierende andere Vorträge gesehen und können zu diesen Fragen stellen</li> </ul>					
<b>Inhalt:</b> In dem Seminar wird ein aktuelles Thema der Algorithmik besprochen. Dies können zum Beispiel parametrisierte oder verteilte Algorithmen sein, oder Algorithmen mit Anwendungen in der Robotik oder in Geographischen Informationssystemen.					
<b>Lehrformen:</b> Seminarvorträge der Studierenden mit anschließender Diskussion					
<b>Prüfungsformen:</b> Semesterbegleitend; Vortrag von 45-60 min, Ausarbeitung zum Vortrag von max. 10 Seiten					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Erfolgreicher Vortrag und Ausarbeitung					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> Informatik, Mathematik					
<b>Stellenwert der Note für die Endnote:</b> 5 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Maike Buchin</b>					

<b>Fortgeschrittene Themen des Model Checking</b>					
<b>Modul-Nr:</b> 150521	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> 4.-6.. Sem.	<b>Turnus:</b> SoSe 21	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Seminar (2 SWS)			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Veranstaltung "Model Checking"					
<b>Lernziele (learning outcomes):</b> Nach dem erfolgreichen Abschluss des Seminars: <ul style="list-style-type: none"> <li>• haben die Studierenden weiterführende Kenntnisse zum Thema Model Checking erworben</li> <li>• können sich Studierende selbständig mit einem Thema aus dem Bereich des Model Checking auseinandersetzen und das Erlernte in einem Vortrag vorstellen</li> </ul>					
<b>Inhalt:</b> Wie kann die Korrektheit von Software und Hardware formal überprüft werden? Im Model Checking werden Software- und Hardware-Module durch Transitionssysteme formalisiert; gewünschte Eigenschaften mit Hilfe logischer Formalismen formal beschrieben; und mit Hilfe von Algorithmen automatisiert überprüft, ob ein Transitionssystem eine formal spezifizierte Eigenschaft besitzt. In der Veranstaltung Model Checking haben wir die theoretischen Grundlagen des Model Checkings kennengelernt. Insbesondere haben wir die Spezifikationsprachen LTL und CTL eingeführt, ihre Ausdrucksstärke untersucht, und die wichtigsten algorithmischen Ansätze für das Model Checking erarbeitet. In diesem Seminar wollen wir uns mit weiterführenden, aktuellen Themen im Bereich Model Checking beschäftigen.					
<b>Lehrformen:</b> seminaristischer Unterricht; Erstellen einer Seminararbeit					
<b>Prüfungsformen:</b> Semesterbegleitend; Präsentation der eigenen Seminararbeit					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Erfolgreiche Bearbeitung einer Seminararbeit; aktive Beteiligung mit Diskussionen an allen Präsentationen					
<b>Verwendung des Moduls (in anderen Studiengängen):</b> M.Sc. Mathematik					
<b>Stellenwert der Note für die Endnote:</b> 3 / 112					
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  <b>Prof. Dr. Thomas Zeume</b>					

**Sonstige Informationen:**

**Literatur:** Edmund M Clarke, Thomas A Henzinger, Helmut Veith, and Roderick Bloem.  
Handbook of model checking, volume 10. Springer, 2018



<b>Seminar Topics in deep learning for sequence processing</b>					
<b>Modul-Nr:</b> 310528	<b>Credits:</b> 3 CP	<b>Workload:</b> 90 h	<b>Semester:</b> 4.-6.. Sem.	<b>Turnus:</b> Wintersemester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Vertiefungsseminare			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 60 h	<b>Gruppen- größe:</b> 24
<p><b>Teilnahmevoraussetzungen:</b> We expect a solid level of mathematics as taught in the Applied Computer Science Bachelor's. Tools commonly used in machine learning are</p> <ul style="list-style-type: none"> <li>• basic probability theory/statistics (expectations, variance, foundational distributions and densities, markov chains)</li> <li>• linear algebra (matrices, vectors, eigenvalues/eigenvectors)</li> <li>• calculus (functions, derivatives/gradients, simple integrals)</li> </ul> <p>The course material is in English, the course language will be English</p>					
<p><b>Lernziele (learning outcomes):</b> After the successful completion of this course the students</p> <ul style="list-style-type: none"> <li>• understand models for sequence processing including recurrent networks and transformers,</li> <li>• understand core challenges of training these models working with time, and</li> <li>• will be able to contextualize current publication with respect to those challenges.</li> </ul>					
<p><b>Inhalt:</b> Natural language processing, robotics, video processing, stock market forecasting and other similar tasks require models that can deal with sequence data and understand temporal dependencies. Two major classes of models that have been designed to deal with sequence data are recurrent neural networks (RNNs/LSTMs) and transformer architectures. Designing and understanding these models is a very active and diverse area of research. Applications of these models are also widespread. The recent explosion of interest in topics such as language modelling and machine translation is based on advances in these models which includes GPT-3, DALL-E, etc.</p> <p>In this course you'll first understand the fundamentals of recurrent neural networks and transformers that led to these breakthroughs. Then we'll go through and discuss both seminal and recent research papers on these topics to throw light on algorithms and challenges in this field.</p>					
<p><b>Lehrformen:</b> Vertiefungsseminar</p>					

## KAPITEL 9. MASTERSEMINARE

<b>Prüfungsformen:</b> Seminarbeitrag
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Presentation of the contents of a recent publication individually or in groups of two. Active participation in discussions during the course.
<b>Stellenwert der Note für die Endnote:</b> 3 / 112
<b>Modulbeauftragte/r und hauptamtlich Lehrende:</b>  Anand Subramoney
<b>Sonstige Informationen:</b> The allocation of the limited seminar places is done via the Moodle course "Masters Seminar: Topics in deep learning for sequence processing" for the corresponding semester. Registration for the winter semester is possible until 11.10.2021. Please enter your degree program in the comment field. Please also check the website for the course: <a href="https://www.ini.rub.de/teaching/courses/topics_in_deep_learning_for_sequence_processing_winter_term_2021/">https://www.ini.rub.de/teaching/courses/topics_in_deep_learning_for_sequence_processing_winter_term_2021/</a>

<b>Student Conference: Research Project</b>					
<b>Modul-Nr:</b> MISSING	<b>Credits:</b> 6 CP	<b>Workload:</b> 180 h	<b>Semester:</b> 1.-3. Sem.	<b>Turnus:</b> Jedes Semester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b>  a) Lectures  b) Individual project work			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 120 h	<b>Gruppen- größe:</b> 1
<b>Teilnahmevoraussetzungen:</b> Keine					
<p><b>Lernziele (learning outcomes):</b> The students work alone (groups can be possible if the project is complex enough) on a scientific project (e.g., literature study, designing an experiment, implementing a technique) of their choice under supervision. Besides obtaining knowledge on the project topic, during the lectures students will learn about scientific work, research methods, scientific writing, presenting, and related topics. Project topics can stem from any innovative field in software engineering (or computer science as a whole). We will provide some example topics, but the students are encouraged to define their own ones together with their supervisor. During the lectures, students will do a short kick-of presentation to introduce their topics to the group. Besides the lectures, the course requires the students to work independently on their project following the research method they selected, with supervision as needed. Before the final submission, a draft is submitted for which the students receive feedback in the form of two reviews by their peers. In the end, they have to submit a scientific paper (refined based on the reviews) and present the results in front of the course; thus simulating a real scientific conference. After this module students</p> <ul style="list-style-type: none"> <li>• understand and can employ basic research methods</li> <li>• are able to critically reflect on and review scientific publications</li> <li>• know how to write scientific papers and present their results</li> <li>• can plan and execute a research project</li> </ul>					
<p><b>Inhalt:</b> Research Methods:</p> <ul style="list-style-type: none"> <li>• Research strategy and process</li> <li>• Research design (experimental, case study, literature survey)</li> <li>• Qualitative, quantitative, and mixed methods</li> <li>• Scientific guidelines</li> <li>• Scientific publication systems (conferences, journals, workshops)</li> </ul>					

## KAPITEL 9. MASTERSEMINARE

Academic Writing: \* Scientific writing \* Style, citations, paraphrasing, punctuation, literature management \* Scientific presentation \* Evaluation of scientific work (peer reviews) \* Discussion/defense of own results

### **Lehrformen:**

- Lectures (scientific writing, research methods)
- Seminar/individual project (working on and presenting a scientific contribution as a paper and presentation)
- Online/hybrid due to COIVD-19

### **Prüfungsformen:**

- Paper (8-10 pages in defined template)
- Oral exam (final presentation of the paper in front of the course in the style of a conference)
- 2 reviews for draft papers of other students (~0.5 pages each)

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

- Submitted and passed/graded paper
- Submission of the 2 reviews
- 2 presentations (kick-off, oral exam)

**Stellenwert der Note für die Endnote: 6 / 112**

**Modulbeauftragte/r und hauptamtlich Lehrende:**

# Studienprojekt

<b>Master-Studienprojekt</b>					
<b>Modul-Nr:</b> N/A	<b>Credits:</b> 10 CP	<b>Workload:</b> 300 h	<b>Semester:</b> 1.-3.. Sem.	<b>Turnus:</b> Jedes Semester	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Praktikum/Projektarbeit			<b>Kontaktzeit:</b> 2 SWS	<b>Selbststudium:</b> 270 h	<b>Gruppen- größe:</b> beliebig
<p><b>Teilnahmevoraussetzungen:</b> Keine</p> <p><b>Empfohlene Vorkenntnisse:</b> Bitte entnehmen Sie die projektspezifischen Vorkenntnisse dem jeweiligen Studienprojekt.</p>					
<p><b>Lernziele (learning outcomes):</b> Ziele des Studienprojekts sind:</p> <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> <p><b>Vermittelte Kompetenzen:</b> Kernkompetenzen:</p> <ul style="list-style-type: none"> <li>• Kritikfähigkeit</li> <li>• Projekt- und Zeitmanagement</li> <li>• Präsentation wissenschaftlicher Ergebnisse</li> </ul>					

## KAPITEL 10. STUDIENPROJEKT

- Teamarbeit und Teamfähigkeit
- Fächerübergreifendes Denken und Arbeiten

### **Inhalt:**

Im Rahmen des Studienprojekts soll eine Aufgabe aus Bereichen der Angewandten Informatik in Teamarbeit unter Anleitung eines Betreuers gelöst werden.

Organisation und Ablauf: Das Studienprojekt muss vor Beginn beim Prüfungsamt angemeldet werden. Bitte beachten Sie, die hierfür festgelegten Fristen unter Sonstiges. In begründeten Fällen (Industriekooperation etc.) können Studienprojekte auch außerhalb dieses Verfahrens durchgeführt werden, wenn Sie durch den Prüfungsausschuss AI beantragt und genehmigt wurden.

Durchführungszeitraum: Studienprojekte starten mit Beginn der Vorlesungszeit und enden spätestens am 31. Juli (Sommersemester) bzw. am 15. Februar (Wintersemester) durch eine Abschlussdokumentation. Die termingerechte Abgabe und positive Bewertung durch den Dozenten sind Voraussetzung für die Anerkennung der Leistungspunkte.

### **Lehrformen:**

Praktikum/Projektarbeit

### **Prüfungsformen:**

Projektarbeit

### **Voraussetzungen für die Vergabe von Kreditpunkten:**

Aktive Teilnahme am Projekt und Bewertung des Abschlussberichts mit min. 50%

**Stellenwert der Note für die Endnote:** 10 / 112

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

**Prof. Dr. Thomas Herrmann**  
Dozenten der RUB

# Freie Wahlfächer

<b>Freie Wahlfächer</b>					
<b>Modul-Nr:</b> N/A	<b>Credits:</b> Min. 8 CP	<b>Workload:</b> 240 h	<b>Semester:</b> 1.-4.. Sem.	<b>Turnus:</b> Winter- und Sommersemes- ter	<b>Dauer:</b> 1 Semester
<b>Lehrveranstaltungen:</b> Frei wählbar			<b>Kontaktzeit:</b> Je nach Lehr- veranstaltung	<b>Selbststudium:</b> h	<b>Gruppen- größe:</b> beliebig
<b>Teilnahmevoraussetzungen:</b> Keine					
<b>Empfohlene Vorkenntnisse:</b> Abhängig von der gewählten Veranstaltung					
<b>Lernziele (learning outcomes):</b> Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den freien Wahlfächern.					
<b>Vermittelte Kompetenzen:</b> Kernkompetenzen: <ul style="list-style-type: none"><li>• Fächerübergreifendes Denken und Arbeiten</li></ul>					
<b>Inhalt:</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> Frei wählbar					
<b>Prüfungsformen:</b> Abhängig von der gewählten Veranstaltung					
<b>Voraussetzungen für die Vergabe von Kreditpunkten:</b> Bestandene Modulabschlussklausur					
<b>Stellenwert der Note für die Endnote:</b> Min. 8 / 112					

**Modulbeauftragte/r und hauptamtlich Lehrende:**

**Prof. Dr. Laurenz Wiskott**

Dozenten der RUB

**Sonstige Informationen:**

Aus der Fakultät für Wirtschaftswissenschaften sind nur die folgenden Veranstaltungen als freie Wahlfächer für die AI geöffnet:

- Finanzierung und Investition
- Grundlagen der Businessplanerstellung
- Grundlagen der Existenzgründung
- Grundlagen der Makroökonomie
- Grundlagen der Mikroökonomie
- Grundlagen des Wirtschaftsrechts
- Jahresabschluss
- Märkte und Unternehmungen (sofern nicht bereits im Bachelor absolviert)
- Marktorientierte Unternehmensführung (sofern nicht bereits im Bachelor absolviert)

Der Besuch weiterer Veranstaltungen ist nur nach Absprache mit dem Dozenten und dem Prüfungsamt AI möglich.