

**Bachelor 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.akafoc.de/inklusion/>

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

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

Studiengangsziele:

Ziel des Bachelorstudiums ist ein wissenschaftlich fundiertes, grundlagenorientiertes Studium, das eine breite und in ausgewählten Teilgebieten vertiefte Basis fachlichen Wissens sowie eine umfassende, auch anwendungsorientierte, Methodenkompetenz vermittelt. Die Vermittlung analytischer, kreativer und gestalterischer Fähigkeiten, aber auch anwendungsbezogener Fertigkeiten im Umgang mit modernen Hard- und Softwaresystemen, unter der Maßgabe, innovative informatische Problemlösungskonzepte neu oder weiter zu entwickeln sowie komplexe Anwendungssysteme mit polydisziplinärer Ausrichtung zu schaffen, sind oberstes Ziel des Studiums. Damit eröffnet der Bachelorstudiengang den Berufszugang. Außerdem vermittelt er die wissenschaftlichen Grundlagen für ein Masterstudium.

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,?). Während des Studiums müssen Pflichtmodule in Informatik, Mathematik und Betriebswirtschaftslehre erfolgreich abgeschlossen werden. Diese vermitteln wichtiges Grundlagenwissen, welches für das weitere Studium und/oder den späteren Beruf essentiell ist. Der Pflichtbereich macht etwa 2/3 des Studiums aus. In allen anderen Modulen können die Studierenden durch zahlreiche Wahlmöglichkeiten ein individuelles Studienprofil entwickeln.

Prüfungsformen

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

Studienplan Bachelor Angewandte Informatik PO 13

Nr	Modul	Lehrveranstaltungen	Mind. Umfang Modul (LP)	Semester der Modulprüfung	autom. Anmeldung	autom. Wiederanmeldung	Bewertung
Pflichtmodule Mathematik							
1	Höhere Mathematik I	Höhere Mathematik I	9	1	Ja	Ja	benotet
2	Statistik und Wahrscheinlichkeitsrechnung	Statistik und Wahrscheinlichkeitsrechnung	5	1	Ja	Ja	benotet
3	Höhere Mathematik II	Höhere Mathematik II	9	2	Ja	Ja	benotet
4	Diskrete Mathematik I	Diskrete Mathematik I	9	3	Nein	Ja	benotet
Pflichtmodule Informatik							
5	Informatik I	Informatik I	5	1	Ja	Ja	benotet
6	Programmieren in C	Programmieren in C	3	1	Ja	Ja	benotet
7	Computernetze	Computernetze	5	2	Ja	Ja	benotet
8	Informatik II	Informatik II	5	2	Ja	Ja	benotet
9	Objektorientierte Modellierung	Objektorientierte Modellierung	5	2	Ja	Ja	benotet
10	Rechnerarchitektur	Rechnerarchitektur	5	3	Nein	Ja	benotet
11	Theoretische Informatik	Theoretische Informatik	9	3	Nein	Ja	benotet
12	Datenstrukturen	Bis WS 20/21: Datenstrukturen ab SS 21: Algorithmenparadigmen	9	4	Nein	Ja	benotet
13	Software-Engineering	Software-Engineering	5	4	Nein	Ja	benotet
14	Web-Engineering	Web-Engineering	5	4	Nein	Ja	benotet
15	Betriebssysteme	Betriebssysteme	5	4	Nein	Ja	benotet
16	Datenschutz	Datenschutz	4	5	Nein	Ja	benotet
17	Datenbanksysteme	Datenbanksysteme	9	5	Nein	Ja	benotet
Pflichtmodule Betriebswirtschaftslehre							
18	Wirtschaftlichkeitsanalyse	Wirtschaftlichkeitsanalyse	5	1	Ja	Ja	benotet
19	Einführung Management Science	Einführung Management Science	6	2	Ja	Ja	benotet
Wahlfächer							
20	Nichttechnische Wahlmodule	Nichttechnische Wahlmodule*	5	1-6	Nein	Nein	unbenotet
Praktische Fächer							
21	Programmierpraktikum	Programmierpraktikum	3	4	Nein	Ja	benotet
22	Studienprojekt	Projektmanagement, Studienprojekt	10	5	Nein	Nein	benotet
Vertiefungsbereich							
23	Vertiefungsmodule	Vertiefungsmodule **	30	3-6	Nein	Ja	benotet
24	Vertiefungsseminar	Vertiefungsseminar **	3	6	Nein	Nein	benotet
Bachelorarbeit							
25	Bachelorarbeit und Kolloquium	Bachelorarbeit und Kolloquium	12	6	Nein	Nein	benotet
Summe:			180				

* Hier können (nahezu) alle Veranstaltungen des Vorlesungsverzeichnisses der RUB, sowie Veranstaltungen im Rahmen der UAMR gewählt werden, sofern es sich dabei um nichttechnische Fächer handelt. Aus der Fakultät Wirtschaftswissenschaften sind nicht alle Veranstaltungen für die AI geöffnet. Eine entsprechende Auflistung befindet sich im aktuellen Modulhandbuch. Eine Liste mit Empfehlungen finden Sie auf der Homepage des Studienganges.

** Die Liste der wählbaren Vertiefungsseminare und Vertiefungsmodule befinden sich im jeweils aktuellen Modulhandbuch, welches vom Gemeinsam beschließenden Ausschuss verabschiedet wird.

Vertiefungsbereich Bachelor Angewandte Informatik PO13

Lehrveranstaltung	Einheit	Umfang Modul (LP)	Semester	Autom. Anmeldung	Autom. Wiederanmeldung	Bewertung
Vertiefungsseminare						
Computerlinguistik (Angebot siehe Vorlesungsverzeichnis)	Philologie	3	WS/SS	Nein	Nein	benotet
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
Bachelor-Seminar Maschinelles Lernen	INI	3	SS	Nein	Nein	benotet
Seminar Knowledge Graphs	Mathe	3	WS	Nein	Nein	benotet
Seminar Deep Learning	Mathe	3	WS	Nein	Nein	benotet
Vertiefungsmodule						
Ingenieurinformatik						
Digitaltechnik (Informatik 3)	ETIT	5	SS	Nein	Ja	benotet
Geometrische Modellierung und Visualisierung	Baulng	6	WS	Nein	Ja	benotet
Grundlagen der Automatisierungstechnik	MB	6	WS	Nein	Ja	benotet
Fertigungsautomatisierung	MB	6	SS	Nein	Ja	benotet
Mensch-Maschine-Interaktion (nicht im WS 21/22)	AW	5	WS	Nein	Ja	benotet
Virtuelle Produktentwicklung	MB	6	WS	Nein	Ja	benotet
Menschenzentrierte Robotik	MB	6	WS/SS	Nein	Ja	benotet
Bioinformatik						
Grundlagen der Bioinformatik	Biologie	5	WS	Nein	Ja	benotet
Methoden der Bioinformatik	Biologie	5	SS	Nein	Ja	benotet
Computerlinguistik						
Einführung in die Linguistik	Philologie	6	WS	Nein	Ja	benotet
Methoden der Computerlinguistik (Symbolische und statistische Verfahren)	Philologie	5	WS	Nein	Ja	benotet
Kryptologie und theoretische Informatik						
Diskrete Mathematik II	Mathe	6	SS	Nein	Ja	benotet
Zahlentheorie	Mathe	9	SS	Nein	Ja	benotet
Einführung in die Kryptographie I	ETIT	5	WS	Nein	Ja	benotet
Einführung in die Kryptographie II	ETIT	5	SS	Nein	Ja	benotet
Kryptographie auf hardwarebasierten Plattformen	ETIT	5	WS	Nein	Ja	benotet
Logik in der Informatik (erst wieder im WS 22/23)	Mathe	5	WS	Nein	Ja	benotet
Automata Theory (nicht im SS 22)	Mathe	5	SS	Nein	Ja	benotet
Neuroinformatik						
Künstliche Neuronale Netze	INI	6	WS	Nein	Ja	benotet
Mathematics for Modeling and Data Analysis	INI	5	SS	Nein	Ja	benotet
Introduction to Computational Neuroscience	INI	6	SS	Nein	Ja	benotet
Introduction to Artificial Intelligence	INI	5	SS	Nein	Ja	benotet
Programmier- und Simulationstechnik						
Agent-based Modeling in Economics and Business (nicht im WS 21/22)	WIWI	5	WS	Nein	Ja	benotet
Game Development	INI	6	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
 WIWI: Fakultät für Wirtschaftswissenschaft

SS: Sommersemester
 WS: Wintersemester
 LP: Leistungspunkte (auch Creditpoints oder CP)

Inhaltsverzeichnis

1	Pflichtmodule Mathematik	1
1.1	Höhere Mathematik I	1
1.2	Statistik und Wahrscheinlichkeitsrechnung	3
1.3	Höhere Mathematik II	5
1.4	Diskrete Mathematik I	7
2	Pflichtmodule Informatik	8
2.1	Programmieren in C	8
2.2	Informatik I	10
2.3	Computernetze	12
2.4	Informatik II	14
2.5	Objektorientierte Modellierung	16
2.6	Rechnerarchitektur	18
2.7	Theoretische Informatik	20
2.8	Datenstrukturen	22
2.9	Software-Engineering	24
2.10	Web-Engineering	26
2.11	Betriebssysteme	28
2.12	Datenschutz	30
2.13	Datenbanksysteme	33
3	Pflichtmodule Betriebswirtschaftlehre	35
3.1	Wirtschaftlichkeitsanalyse	35
3.2	Einführung Management Science	37
4	Nichttechnische Wahlmodule	39
4.1	Nichttechnische Wahlmodule	39
5	Praktische Fächer	41
5.1	Programmierpraktikum	41
5.1.1	Intensivkurs C++	41
5.1.2	Introduction to Python	43
5.1.3	Programmierpraktikum Android	45
5.1.4	Grundlagen Roboterprogrammierung	47
5.1.5	Introduction to CL with Python	49
5.2	Studienprojekt	50
5.2.1	Studienprojekt	50
5.2.2	Projektmanagement	52

6	Bachelorarbeit	54
6.1	Bachelorarbeit	54
7	Ingenieurinformatik	56
7.1	Digitaltechnik (Informatik 3)	56
7.2	Geometrische Modellierung und Visualisierung	59
7.3	Grundlagen der Automatisierungstechnik	61
7.4	Mensch-Maschine-Interaktion	64
7.5	Virtuelle Produktentwicklung	67
7.6	Fertigungsautomatisierung	69
7.7	Menschenzentrierte Robotik	71
8	Bioinformatik	74
8.1	Grundlagen der Bioinformatik	74
8.2	Methoden der Bioinformatik	76
9	Computerlinguistik	78
9.1	Einführung in die Linguistik	78
9.2	Methoden der Computerlinguistik	82
10	Kryptologie	84
10.1	Diskrete Mathematik II	84
10.2	Zahlentheorie	86
10.3	Einführung in die Kryptographie 1	88
10.4	Einführung in die Kryptographie 2	90
10.5	Kryptographie auf hardwarebasierten Plattformen	92
10.6	Logik in der Informatik	94
10.7	Automata theory	96
11	Neuroinformatik	98
11.1	Einfuehrung KI	98
11.2	Computersehen: Einführung	100
11.3	Kuenstliche Neuronale Netze	102
11.4	Mathematics for Modeling and Data Analysis	104
11.5	Introduction Computational Neuroscience	105
12	Programmier- und Simulationstechnik	107
12.1	Game Development	107
12.2	Agent-based modeling in economics and business	109
13	Vertiefungsseminar	111
13.1	Seminar Computerlinguistik	111
13.2	Seminar Ingenieurinformatik	114
13.3	Seminar Kryptologie und Theoretische Informatik	116
13.4	Seminar Maschinelles Lernen	118

Pflichtmodule Mathematik

Höhere Mathematik I					
Modul-Nr: 150160 (Übung: 150161)	Credits: 9 CP	Workload: 270 h	Semester: 1. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: V: 4 SWS; Ü: 2	Selbststudium: 180 h	Gruppen- größe: Vorlesung: Unbegrenzt; Übung: 50 pro Gruppe
Teilnahmevoraussetzungen: Keine					
Empfohlene Vorkenntnisse: Mathematische Schulausbildung (Gymnasiale Oberstufe) und die Teilnahme an einem Mathematik- Vorkurs					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls: Die Studierende					
<ul style="list-style-type: none"> • Verstehen und wenden grundlegende mathematische Begriffe und Notationen an • Nutzen und führen die vermittelten mathematischen Methoden aus • Übertragen und wenden zugehörige Lösungsverfahren auch auf praktische Probleme an 					
Vermittelte Kompetenzen:					
Kernkompetenzen:					
<ul style="list-style-type: none"> • Analytisches und logisches Denken 					
Fachspezifische Kompetenzen:					
<ul style="list-style-type: none"> • Mathematik 					
Inhalt:					
Aussagenlogik, Mengen und Abbildungen, Reelle Zahlen und algebraische Strukturen, Komplexe Zahlen, Folgen und Reihen, Stetige Funktionen, Differenzialrechnung (in \mathbb{R}), Inte-					

KAPITEL 1. PFLICHTMODULE MATHEMATIK

Integralrechnung (in \mathbb{R})
Lehrformen: Vortrag der Lehrenden in der Vorlesung (mit zum Teil digitalen Lehrformaten), Gruppenarbeit in den Übungen, Ergänzung der Bearbeitung der Hausaufgaben in Einzel- oder Gruppenarbeit durch digitale Aufgaben, Online-Tests
Prüfungsformen: Klausur (120 Minuten)
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 9 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: PD Dr. Daniela Kacso

Statistik und Wahrscheinlichkeitsrechnung					
Modul-Nr: 070030 (Kolloquium: 070031/070032, Tutorium: 076040)	Credits: 5 CP	Workload: 150 h	Semester: 1. Sem.	Turnus: Jedes Semester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Kolloquium / Tutorium			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: Kolloquium: 60 – 100 Studieren- de, Tutorium: 20 - 40 Studie- rende
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Generische Kompetenzen:</p> <ul style="list-style-type: none"> • Abstraktion und logisches Denken (systemische Kompetenz), • Analyse und Strukturierung von Problemen (systemische K.) <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Auswahl geeigneter Methoden zur Problemlösung (instrumentale K.) • Kontextspezifische (ökonomische) Interpretation und Darstellung von Resultaten (kommunikative K.) <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • verfügen Studierende über grundlegende Methodenkenntnisse im Bereich der Wahrscheinlichkeitsrechnung, • kennen Studierende die Konzepte der Schließenden Statistik 					
<p>Inhalt: Die Veranstaltung besteht aus zwei Teilen. Der erste Teil behandelt die Wahrscheinlichkeitsrechnung. Nach einer kurzen Übersicht über die klassische und die axiomatische Wahrscheinlichkeitsrechnung werden grundlegende Eigenschaften von Zufallsvariablen und für die wirtschaftswissenschaftliche Anwendung wichtigen parametrischen Verteilungen vorgestellt. Anschließend werden Approximationsaussagen, die u.a. auf den Zentralen Grenzwertsätzen beruhen, eingeführt. Im zweiten Teil wird eine Einführung in die Methoden der schließenden Statistik gegeben. Dazu werden die Konzepte von Zufallsstichproben, Punkt- und Intervallschätzung, Signifikanztests und Regressionsanalyse einführend dargestellt. In den vorlesungs-</p>					

KAPITEL 1. PFLICHTMODULE MATHEMATIK

begleitenden Kolloquien und Tutorien werden die Anwendungen der Methoden anhand von Fallbeispielen und Aufgaben eingeübt.

Lehrformen:

Hörsaalvorlesung mit Medienunterstützung, Kolloquien als seminaristischer Unterricht, Tutorien als seminaristischer Unterricht oder Inverted Classroom, zusätzlich jeweils Selbststudium mit ergänzend bereitgestellten Materialien und Aufgaben

Prüfungsformen:

Modulklausur (90 Minuten)

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Verwendung des Moduls (in anderen Studiengängen): Studiengang Management and Economics (Pflichtmodul Grundlagenphase), 2-Fach Bachelor Wirtschaftswissenschaft mit Ausrichtung BWL (Pflichtmodul Grundlagenphase), 2-Fach Bachelor Wirtschaftswissenschaft mit Ausrichtung VWL (Pflichtmodul Grundlagenphase), Nebenfach Wirtschaftswissenschaft, Studierende anderer Fakultäten (über das Zentrum für Ökonomische Bildung)

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Dr. Markus Pape

Dr. Benno Hildebrandt, Markus Pape

Sonstige Informationen:

Die Modulbezeichnung an der Fakultät für Wirtschaftswissenschaft lautet „Statistik II“. Das Modul „Statistik I“ in den wirtschaftswissenschaftlichen Bachelorstudiengängen beschäftigt sich u.a. mit deskriptiver Statistik, elementarer Zeitreihenanalyse und Indexzahlen.

Die vorlesungsbegleitende Übung wird in zwei Formaten angeboten. Kolloquien werden in größeren Gruppen abgehalten, Tutorien in kleineren Gruppen. Kolloquien werden üblicherweise von Mitarbeiterinnen und Mitarbeitern des Lehrstuhls gehalten, in den Tutorien kommen Studierende höherer Semester zum Einsatz.

Höhere Mathematik II					
Modul-Nr: 150162 (Übung: 150163)	Credits: 9 CP	Workload: 270 h	Semester: 2. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: V: 4 SWS; Ü: 2	Selbststudium: 180 h	Gruppen- größe: Vorlesung: Unbegrenzt; Übung: 50 pro Gruppe
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Höhere Mathematik I					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls: Die Studierende <ul style="list-style-type: none"> • Verstehen und wenden grundlegende mathematische Begriffe und Notationen an • Sind in der Lage die Themen zu erläutern und Einsatzmöglichkeiten zu benennen • Können Formalismen und Verfahren auswählen und ausführen, die erzielten Ergebnisse interpretieren • Übertragen und wenden zugehörige Lösungsverfahren auch auf praktische Probleme an Vermittelte Kompetenzen: Kernkompetenzen: <ul style="list-style-type: none"> • Analytisches und logisches Denken Fachspezifische Kompetenzen: <ul style="list-style-type: none"> • Mathematik 					
Inhalt: Potenzreihen und Fourierreihen, Vektorräume, Lineare Gleichungssysteme, Matrizen, Determinanten, Eigenwerte und Eigenvektoren, Differenzialrechnung (in \mathbb{R}^n), Gewöhnliche Differenzialgleichungen					
Lehrformen: Vortrag der Lehrenden in der Vorlesung (mit zum Teil digitalen Lehrformaten), Gruppenarbeit in den Übungen, Ergänzung der Bearbeitung der Hausaufgaben in Einzel- oder Gruppenarbeit durch digitale Aufgaben, Online-Tests					
Prüfungsformen: Klausur (120 Minuten)					

KAPITEL 1. PFLICHTMODULE MATHEMATIK

Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 9 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: PD Dr. Daniela Kacso

KAPITEL 1. PFLICHTMODULE MATHEMATIK

Diskrete Mathematik I					
Modul-Nr: 150308 (Übung: 150309)	Credits: 9 CP	Workload: 270 h	Semester: 3. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung Diskrete Mathematik I b) Übungen zu a)			Kontaktzeit: V: 4 SWS; Ü 2	Selbststudium: 180 h	Gruppen- größe: Übung: 40 pro Gruppe; Vorlesung: unbegrenzt
Teilnahmevoraussetzungen: Keine					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls					
<ul style="list-style-type: none"> • haben die Studierenden abstrakte diskrete Strukturen kennengelernt und können diese sicher erläutern • können sie konkrete Problemstellungen mit den kennengelernten Strukturen modellieren • haben sie grundlegende algorithmische Techniken kennen- und anwenden gelernt und können diese Algorithmen analysieren • können sie logische Zusammenhänge zwischen den verschiedenen kennengelernten Konzepten erkennen und versteckte Anwendungsmöglichkeiten finden 					
Inhalt: Kombinatorik, Abzählprobleme; Grpahentheorie: Graphenexploration und weitere ausgesuchte Graphenprobleme; Grundkenntnisse in elementarer Zahlentheorie mit kryptographischen Anwendungen, Designtechniken für effiziente Algorithmen, Aufstellen und Lösen von Rekursionsgleichungen; grundlegende algebraische Strukturen mit Anwendungen auf symmetrische Zählprobleme und fehlerkorrigierende Codes					
Lehrformen: Vortrag der Lehrenden in der Vorlesung und Kleingruppenarbeit in den Übungen, Bearbeitung wöchentlicher Hausaufgaben einzeln oder in Kleingruppen					
Prüfungsformen: Klausur 180 Minuten					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					
Stellenwert der Note für die Endnote: 9 / 175					
Modulbeauftragte/r und hauptamtlich Lehrende: PD Dr. Björn Schuster					

Pflichtmodule Informatik

Programmieren in C					
Modul-Nr: 149872	Credits: 3 CP	Workload: 90 h	Semester: 1. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 3 SWS	Selbststudium: 45 h	Gruppen- größe: ca. 500
Teilnahmevoraussetzungen: Keine					
Lernziele (learning outcomes): Die Studierenden beherrschen die grundlegenden Sprachkonstrukte von C mit Betonung der prozeduralen Betrachtungsweise und haben ein Verständnis für die Sicherheitsproblematik von C. Vermittelte Kompetenzen: Fachspezifische Kompetenzen: <ul style="list-style-type: none">• Software-Entwurfsmethoden Technologien: <ul style="list-style-type: none">• C• Matlab					
Inhalt: <ul style="list-style-type: none">• Verfahren der Strukturierten Programmierung• Einführung in die Programmiersprache C (C90/C99/C11)• Elementare Sprachkonstrukte (Standard-Datentypen, Ausdrücke, Kontrollstrukturen)• Prozedurale Betrachtungsweise (Funktionen und Programmstrukturen)• Klassische Datenstrukturen (Arrays, Verbunde) und Zeiger• Dynamische Datenstrukturen					

KAPITEL 2. PFLICHTMODULE INFORMATIK

<ul style="list-style-type: none">• Sicherheitsproblematik
Lehrformen: Vorlesung (+Übung)
Prüfungsformen: Klausur
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 3 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Markus Dürmuth Prof. Dr. Markus Dürmuth, André Feiler
Sonstige Informationen: Die Veranstaltung wurde letztmalig im WS 2020/2021 angeboten Literatur: Skript

Informatik 1 - Programmierung					
Modul-Nr: 141328	Credits: 5 CP	Workload: 150 h	Semester: 1. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen:			Kontaktzeit: 3 SWS	Selbststudium: 105 h	Gruppen- größe: beliebig, zuletzt über 500 Teil- nehmer
<ul style="list-style-type: none"> • Vorlesung (2 SWS) • Übung (1 SWS) 					
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Digitalisierung: Hausaufgaben werden automatisiert korrigiert und bepunktet. Es werden kurze Lernabschnitte im flipped-classroom Format eingefügt. Zur Aktivierung gibt es kurze Abstimmungen im Hörsaal. Die Vorlesung wird aufgezeichnet und als Video-Stream zur Verfügung gestellt. Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • kennen die Teilnehmer die wichtigsten Konzepte imperativer und objektorientierter Programmierung, • können die Teilnehmer eigene Programme schreiben, • verstehen die Teilnehmer erste Begriffe der Informatik wie etwa Laufzeit in Landau-Notation. 					
<p>Inhalt: Imperative und objektorientierte Programmierung, sowie erste Grundbegriffe der Informatik</p>					
<p>Lehrformen: Vorlesung und Übung</p>					
<p>Prüfungsformen: Klausur 120 Minuten, Wiederholungsklausur nach 6 Monaten</p>					
<p>Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur</p>					
<p>Verwendung des Moduls (in anderen Studiengängen): Bachelor Elektrotechnik Bachelor IT-Sicherheit</p>					
<p>Stellenwert der Note für die Endnote: 5 / 175</p>					
<p>Modulbeauftragte/r und hauptamtlich Lehrende:</p> <p>Prof. Dr. Tobias Glasmachers Prof. Dr. Tobias Glasmachers, Florian Fricke</p>					

Sonstige Informationen:

Ab WS 20/21 wird die Veranstaltung mit 8 CP angeboten

Computernetze					
Modul-Nr: 141250	Credits: 5 CP	Workload: 150 h	Semester: 2. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Computernetze			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 400 Studie- rende pro Vorle- sung/Übung
Teilnahmevoraussetzungen: Keine					
Empfohlene Vorkenntnisse: Elementare Grundkenntnisse der Informatik					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls					
<ul style="list-style-type: none"> • kennen Studierende die wichtigsten Standards, die das heutige Internet verwendet. • kennen Studierende grundlegende Angriffskonzepte auf Computernetzwerke • verstehen Studierende den Zusammenhang zwischen den einzelnen Schichten eines Computernetzwerks und der darin enthaltenen Protokolle • können Studierende die wichtigsten Netzwerktools für Analysezwecke anwenden 					
Inhalt:					
<p>Die Vorlesung gibt eine Einführung in grundlegenden Protokolle und Anwendungen von Computernetzen. Der Schwerpunkt der Vorlesung liegt auf Standardprotokollen und -Algorithmen, wie sie in modernen Computernetzwerken (zum Beispiel im Internet) eingesetzt werden.</p> <p>Anhand eines Schichtenmodells werden die wichtigsten Grundlagen nach dem Top-Down Ansatz vorgestellt und analysiert. Dazu gehören zum Beispiel auf der obersten Schicht DNS und HTTPS im Application Layer; TCP und UDP im Transport Layer; IPv4/IPv6 und Routing Algorithmen im Network Layer; sowie MAC und ARP im untersten Link Layer. Neben der reinen Funktionsweise dieser Standards werden Sicherheitsaspekte auf allen Schichten betrachtet.</p> <p>Ergänzend zur Vorlesung werden Übungsaufgaben über die eLearning Plattform Moodle gestellt und in der Übungsstunde besprochen. Weiterhin wird in jeder Übung ein "Tool der Woche" vorgestellt. Dabei handelt es sich jeweils um eine spezielle Software, die man als "Netzwerker" unbedingt kennen sollte (z.B. traceroute, nmap, ...). Alle besprochenen Tools sind frei verfügbar und werden den Studenten als eine Lernplattform (virtuelle Maschine) zur Verfügung gestellt.</p> <p>Als Primärliteratur wird "Computernetzwerke: Der Top-Down Ansatz" von Kurose und Ross (Pearson Verlag) verwendet.</p>					
Lehrformen:					
Moodle-Unterstützte Hausaufgaben mit praxisnahen, computerunterstützten Übungen. Tool-					

KAPITEL 2. PFLICHTMODULE INFORMATIK

der-Woche: Vorstellung, Einarbeitung, und Verwendung von Netzwerkrelevanten Computer-analysertools
Prüfungsformen: schriftliche Modulabschlussprüfung von 120 min
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Verwendung des Moduls (in anderen Studiengängen): Studiengang Bachelor IT-Sicherheit (ET/IT), Studiengang Bachelor Angewandte Informatik
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Christian Mainka
Sonstige Informationen: Literatur: 1. "Computernetzwerke", Kurose und Ross, Pearson. 2. "Computernetzwerke", Tanenbaum und Wetherall, Pearson.

Informatik II					
Modul-Nr: 150322 Übung (150323)	Credits: 5 CP	Workload: 150 h	Semester: 2. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 500
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Inhalte der Vorlesung Informatik I</p>					
<p>Lernziele (learning outcomes): Die Studierenden erhalten einen systematischen Überblick über Prinzipien, Methoden, Konzepte und Notationen von verschiedenen Algorithmen und Datenstrukturen. Dieses Wissen - verbunden mit den praktischen Übungen am Computer - befähigt die Studierenden, effiziente Programme problemgerecht zu entwickeln, zu analysieren, zu überprüfen und in die Programmiersprache C++ zu transformieren, zu übersetzen und bzw. darin auszuführen.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Algorithmik • Graphentheorie • Programmieren <p>Technologien:</p> <ul style="list-style-type: none"> • C++ 					
<p>Inhalt: Die Vorlesung besteht aus vier größeren Blöcken, die wie folgt aufgeteilt sind: Im ersten Block werden Grundbegriffe eingeführt, der Fokus liegt auf dem Vergleich und der Bewertung von Algorithmen. Im zweiten Block werden klassische Sortalgorithmen (z: B. insertion sort, mergesort und quicksort) vorgestellt. Klassische Suchalgorithmen wie binäre Suche oder verschiedene Arten von Baumstrukturen werden im dritten Block vorgestellt. Im letzten Block wird ein Überblick über Graphalgorithmen und Operationen auf Strings gegeben. Begleitet wird die Vorlesung von Übungen, in denen die vorgestellten Konzepte und Techniken praktisch ausprobiert werden sollen.</p>					
<p>Lehrformen: Vorlesung (+Übung)</p>					

KAPITEL 2. PFLICHTMODULE INFORMATIK

Prüfungsformen: Klausur
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Tim Güneysu
Sonstige Informationen: Alle Materialien sind per Moodle verfügbar, bitte registrieren Sie sich dort für den Kurs. Literatur: Die Veranstaltung basiert auf dem Buch "Algorithms" von Sedgewick und Wayne (siehe http://algs4.cs.princeton.edu/).

Objektorientierte Modellierung					
Modul-Nr: 128047 (Übung: 128048)	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Objektorientierte Modellierung			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Einführung in die Informatik und Programmierung					
<p>Lernziele (learning outcomes): Globales Ziel dieser Veranstaltung ist es, einen systematischen Überblick über Prinzipien, Methoden, Konzepte und Notationen der Objektorientierte Modellierung zu geben. Dieses Wissen – verbunden mit den praktischen Übungen am Computersystem – soll den Studierenden befähigen, nach internationalen Standards konzipierte, effiziente Programme problemgerecht zu entwickeln, zu analysieren und zu überprüfen. Dazu gehört das Beschreiben von Sachverhalten mittels der UML (Unified Modeling Language) und das Arbeiten mit der Programmiersprache Java. Besonders mit fortschreitender Digitalisierung ist es zentral, Programmstrukturen entwickeln zu können, welche einfach zu warten sind und es ermöglichen, umfangreiche Teams an einem Softwareprojekt zu beschäftigen.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • Können Studierende in der Programmiersprache Java programmieren. • Können Studierende Objektorientierte Methoden auf Programmiersprachen abstrahieren. • Können Studierende geläufige Objektorientierte Muster erkennen, anwenden und programmieren. • Sind Studierende in der Lage, umfassende Programmstrukturen eigenständig zu entwerfen, sodass diese übersichtlich und wartbar sind. • Können Studierende grafische Benutzeroberflächen in Java entwerfen. 					
<p>Inhalt: Basiskonzepte der Objektorientierung - Klassen und Objekte - Generalisierung - Vererbung und Schnittstellen - Assoziationen - Generische Datentypen und Container - Unified Modeling Language GUI-Programmierung - Ereignisverarbeitung - Model-View-Controller-Prinzip - Dialog- und E/A-Gestaltung</p>					
Lehrformen: Vorlesungen, digitale und vor Ort betreute Übungen					
Prüfungsformen:					

KAPITEL 2. PFLICHTMODULE INFORMATIK

Schriftliche Klausur im Umfang von 120 Minuten
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Markus König
Sonstige Informationen: Literatur: 1. Balzert, Helmut, Priemer, Jürgen "Java 6: Anwendungen programmieren, 2. Auflage", W3l, 2010 2. Balzert, Helmut "Java:Objektorientiert programmieren, 2. Auflage", W3l, 2010

Rechnerarchitektur					
Modul-Nr: 141140	Credits: 5 CP	Workload: 150 h	Semester: 3. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 300-400
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Digitaltechnik, Programmiersprachen, Eingebettete Prozessoren</p>					
<p>Lernziele (learning outcomes): Die Studierenden kennen Zusammenhänge und haben Detailkenntnisse bezüglich der Komponenten und der Funktionsweise moderner Computersysteme. Dies schließt neben dem Prozessor auch das Speichersystem und die Schnittstellen zu weiteren Systemkomponenten ein. Auf der Basis dieser Kenntnisse sind die Studierenden in der Lage Computersysteme und deren Komponenten bezüglich verschiedener Metriken, wie z.B. Energieverbrauch, Rechenleistung, Speicherperformance etc. auf deren Eignung für eine bestimmte Aufgabe zu bewerten. Weiterhin haben die Teilnehmer dieser Veranstaltung die grundsätzliche Arbeitsweise und den prinzipiellen Aufbau von Prozessoren auf der Ebene der Mikroarchitektur verstanden und sind in der Lage, den Einfluss von Architekturmerkmalen, wie z.B. Pipelining oder Outof-Order-Execution, auf die Befehlsausführung zu analysieren.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten • Kritikfähigkeit • Literaturrecherche und Dokumentation • Selbständiges Lernen und Arbeiten • Teamarbeit und Teamfähigkeit • Wissenschaftliches Arbeiten und Schreiben <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Logik und Schaltungen 					
<p>Inhalt: Die Veranstaltung Rechnerarchitektur befasst sich mit dem Aufbau und der Funktion moderner Prozessoren und Computersysteme. Ausgehend von grundlegenden Computerstrukturen wie der Von-Neumann- und der Harvard-Architektur werden der Aufbau, die Klassifi-</p>					

KAPITEL 2. PFLICHTMODULE INFORMATIK

zierung und die technische Realisierung von Rechnersystemen dargestellt. Hierbei wird die Programmierung auf Assemblerebene sowie die Verarbeitung von Programmen durch einen Prozessor erläutert. Darauf aufbauend folgen Methoden zu Leistungsbewertung von Prozessoren auf der Basis von standardisierten Benchmarks und verschiedene Metriken, um die Ergebnisse einordnen zu können. Der inhaltliche Schwerpunkt der Vorlesung stellt die tiefgehende Analyse der Mikroarchitekturebene eines Prozessors dar, wobei sowohl der Datenpfad als auch das Steuerwerk im Rahmen der Vorlesung schrittweise entwickelt und erläutert werden. Auf der Basis des in der Vorlesung vorgestellten Prozessors werden dann moderne Verfahren zur Leistungssteigerung und deren Einsatzgebiete vorgestellt. Neben dem eigentlichen Prozessor wird auch das Speichersystem moderner Computer und verschiedene Schnittstellen zu internen und externen Komponenten des Computersystems behandelt. Alle Themen werden mit aktuellen Beispielen aus verschiedenen Bereichen der Technik erläutert, so dass neben dem im Detail vorgestellten Beispielprozessor mit MIPS Architektur auch moderne Hochleistungsprozessoren mit x86-64 ISA, Prozessoren für eingebettete Systeme auf Basis der ARM-Architektur, extrem energiesparende Prozessoren auf Basis des MSP430, wie sie zum Beispiel in IoT-Geräten zum Einsatz kommen, und anwendungsspezifische Spezialprozessoren auf Basis der Tensilica Xtensa Plattform vorgestellt werden.

Lehrformen:

Vorlesung (Übung)

Prüfungsformen:

Klausurarbeit (120 Minuten)

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:**Studiendekan ETIT**

Herr M. Sc. Florian Fricke

Sonstige Informationen:**Literatur:**

1. Tanenbaum, Andrew S. "Computerarchitektur. Strukturen - Konzepte - Grundlagen", Pearson, 2006
2. Tanenbaum, Andrew S. "Modern Operating Systems", Pearson, 2009
3. Tanenbaum, Andrew S. "Moderne Betriebssysteme", Pearson, 2009
4. Patterson, David A., Hennessy, John L., Bode, Arndt "Rechnerorganisation und -entwurf", Spektrum Akademischer Verlag, 2005
5. Tanenbaum, Andrew S. "Structured Computer Organization", Prentice Hall, 2005

Theoretische Informatik					
Modul-Nr: 150240 (Übung: 150241)	Credits: 9 CP	Workload: 270 h	Semester: 3. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: 6 SWS	Selbststudium: 180 h	Gruppen- größe: Ca. 120 Stu- dierende in der Vorlesung, Ca 30 Studierende in der Übung
Teilnahmevoraussetzungen: Elementare Grundkenntnisse in der Informatik.					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende grundlegende Arten von Automaten und formalen Sprachen • können Studierende formale Sprachen in die Chomsky-Hierarchie einordnen und einen erkennenden Automaten dafür angeben • verstehen Studierende den Begriff der Berechenbarkeit • können Studierende für einfache Funktionen Turingmaschinen angeben, die diese be- rechnen • verstehen Studierende grundlegenden Konzepte der algorithmischen Komplexität von Problemen, insbesondere die Komplexitätsklassen P und NP, sowie den Begriff der NP- Schwerheit 					
Inhalt: Die Vorlesung liefert eine Einführung in die Theorie der Grammatiken (insbesondere die Chomsky-Hierarchie) und Automaten (endlicher Automat, Kellerautomat, Turing-Maschine). Sie gibt ferner einen Einblick in die Berechenbarkeits- und NP-Vollständigkeitstheorie. Hierbei geht es um die Frage, welche Rechenprobleme (überhaupt bzw. mit vertretbarem Aufwand) gelöst werden können. Dabei zeigt sich, dass es inhärent schwere Probleme gibt, die von Rechnern nicht zufriedenstellend gelöst werden können.					
Lehrformen: Vorlesung (als Tafelvortrag) und Übungen					
Prüfungsformen: Klausur über 180 Minuten					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					

Verwendung des Moduls (in anderen Studiengängen): Angewandte Informatik, Informatik, Mathematik, IT-Sicherheit
Stellenwert der Note für die Endnote: 9 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Maike Buchin
Sonstige Informationen: Literatur: Die Vorlesung orientiert sich hauptsächlich an dem Buch "Theoretische Informatik - kurzgefasst" von Uwe Schöning (HTb, 2001). Zur Vertiefung des Stoffes kann sowohl das Buch "Introduction to Automata Theory, Languages, and Computation" von John E. Hopcroft, Rajeev Motwani und Jeffrey D. Ullman (Addison Wesley, 2000) als auch "Introduction to the Theory of Computation" von M. Sipser (PWS Publishing Company, 1997) dienen.

Datenstrukturen					
Modul-Nr: 150322 (Übung: 150323)	Credits: 9 CP	Workload: 270 h	Semester: 4. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: 6 SWS	Selbststudium: 180 h	Gruppen- größe: Ca. 120 Stu- dierende in der Vorlesung, ca. 30 Studierende in der Übung
Teilnahmevoraussetzungen: Die Kenntnis einer höheren Programmiersprache ist hilfreich, aber nicht im engen Sinne erforderlich.					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende grundlegende Datenstrukturen und Algorithmenparadigmen • können Studierende die Korrektheit eines Algorithmus formal beweisen • können Studierende die Laufzeit und den Speicherbedarf von Algorithmen und Datenstrukturen analysieren und bewerten • können Studierende Algorithmen und Datenstrukturen für spezifische Probleme entwickeln 					
Inhalt: Die Vorlesung gibt einen systematischen Überblick über den Entwurf und die Analyse von Algorithmen und Datenstrukturen. Dazu werden zunächst grundlegenden Methoden der Analyse (insbesondere Korrektheit, Laufzeit und Speicherbedarf) von Algorithmen vorgestellt. Anschließend sehen wir einige Algorithmen zum Sortieren und Suchen. Ebenfalls werden verschiedene grundlegende Datenstrukturen (Listen, Felder, Suchbäume und Heaps) vorgestellt. Schließlich betrachten wir Graphen: ihre Darstellung und diverse Algorithmen auf Graphen (Durchläufe, Kürzeste Wege, Minimale Spannbäume). In den Übungen lernen die Studierende sowohl die theoretische Analyse von Algorithmen und Datenstrukturen als auch deren praktische Umsetzung in einer modernen Programmiersprache (z.B. Python oder C++).					
Lehrformen: Vorlesung (als Folien- oder Tafelvortrag) und Übung					
Prüfungsformen: Klausur über 180 Minuten					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					

Verwendung des Moduls (in anderen Studiengängen): Angewandte Informatik, Informatik, Mathematik, IT-Sicherheit
Stellenwert der Note für die Endnote: 9 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Maike Buchin
Sonstige Informationen: Ab SoSe 2021 muss "Algorithmenparadigmen" als Ersatzveranstaltung genutzt werden! Informationen dazu im Vorlesungsverzeichnis Veranstaltungsnummer 150340 Literatur: 1. Güting, Ralf Hartmut und Dieker, Stefan. Datenstrukturen und Algorithmen. Teubner. 2. Aho, Hopcroft und Ullman. The design and analysis of computer algorithms. Addison-Wesley.

Software-Engineering					
Modul-Nr: 260085	Credits: 5 CP	Workload: 150 h	Semester: 4. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übungen			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: 15 pro Übungs- gruppe. 3-4 Studierende pro Fallbeispielauf- gabe
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Objektorientierte Modellierung, Programmierkenntnisse.					
<p>Lernziele (learning outcomes): Die Studierenden</p> <ul style="list-style-type: none"> • können Grundlagen, Aktivitäten und Methoden des Requirements-Engineering und von Softwareentwicklungsprozessen erläutern und einordnen • verfügen über vertiefte Kenntnisse über ausgewählte Aspekte des Softwareentwicklungsprozesses • verfügen über Grundkenntnisse zum Thema Softwarequalität • kennen und verstehen die grundsätzlichen Ziele und Verantwortlichkeiten im Software-Life-Cycle • kennen und verstehen die verschiedenen Aktivitäten innerhalb des Software-Life-Cycles und deren Abhängigkeiten <p>Vermittelte Kompetenzen: Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Entwicklungsprozesse • Software-Entwurfsmethoden <p>Technologien:</p> <ul style="list-style-type: none"> • UML • Eclipse 					
Inhalt:					

KAPITEL 2. PFLICHTMODULE INFORMATIK

Die Studierenden lernen unterschiedliche Formen von (klassischen und agilen) Vorgehensmodellen in der Softwareentwicklung kennen. Sie lernen Methoden der Anforderungserhebung, des Entwurfs und des Testens kennen und setzen diese in einem realen Fallbeispiel selbstständig um.

Die Vorlesung wird von Experten aus der Praxis als Gastdozenten begleitet vertiefende Einblicke in Aufgabenfelder und aktuelle Forschungsthemen zu erhalten.

Lehrformen:

Die Inhalte werden in Vorlesungen vermittelt und in kleinen Übungsgruppen vertieft und eingeübt. Bei letzterem werden alle Methoden und Kenntnisse für die Bearbeitung des Fallbeispiels behandelt. Die Studierenden präsentieren zweimal im Semester ihre Ergebnisse beim Fallbeispiel.

Prüfungsformen:

semesterbegleitend: schriftliche Ausarbeitung von 2 Fallbeispielen und mündliche Überprüfung der Leistungen. Die Endnote setzt sich aus der Note der schriftlichen Abgabe der Phase 1 (40%), der schriftlichen Abgabe der Phase 2 (40%) und dem finalen Vortrag (20%) zusammen.

Voraussetzungen für die Vergabe von Kreditpunkten:

Die CP werden vergeben, wenn eine Mindestnote von 50% erreicht wurde.

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Thomas Herrmann
Marcel Schmittchen Msc.

Sonstige Informationen:

Ab WiSe 2021/22 muss "Software Engineering" (PO 2020) bei Prof. Thorsten Berger als Ersatzveranstaltung genutzt werden! Informationen dazu im Vorlesungsverzeichnis Veranstaltungsnummer 141346.

Literatur:

1. Ian Sommerville: Software Engineering; Addison-Wesley, 2001 (6th edition)
2. Klaus Pohl: Requirements Engineering: Grundlagen, Prinzipien, Techniken, dpunkt.verlag, 2. Aufl., 2008

Web-Engineering					
Modul-Nr: 128968	Credits: 5 CP	Workload: 150 h	Semester: 4. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 100
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Die Entwicklung von Web-Anwendungen und -Services ist zentraler Bestandteil der Digitalisierung. Ziel der Lehrveranstaltung ist die Vermittlung von Grundlagen und bewährten Verfahren in der Web-Entwicklung. Studierende lernen konzeptuelle technologische Bausteine kennen: Transportverfahren, Webseitendarstellung, dynamische Web-Anwendungen und Web-Services. Über das konzeptuelle Verständnis hinaus werden praktische Kompetenzen vermittelt. Dazu werden moderne Werkzeuge der Web-Entwicklung, sowohl server- als auch clientseitig, vorgestellt und in den Übungssitzungen praktisch vertieft. Während der Umsetzung einfacher Web-Anwendungen stehen auch analytische Fähigkeiten im Fokus: Studierende werden befähigt, verschiedene Verfahren in Hinblick auf Performanz und Wartbarkeit zu bewerten. Diese Fähigkeiten sind in der kritischen Planungsphase von Softwareprojekten unerlässlich.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • kennen Studierende gängige Konzepte der Web-Entwicklung in den Aspekten Präsentation, Transport und Bereitstellung von Daten; • beherrschen Studierende grundlegende Fähigkeiten in Webseitendarstellung, dynamischen Web-Anwendungen und modernen Services (Node.js). 					
<p>Inhalt:</p> <ul style="list-style-type: none"> • Einführung in clientseitige Web-Entwicklung: HTML, CSS, JavaScript, Web Components; • Transportverfahren und deren Nutzung: Representational State Transfer (REST), Asynchronous JavaScript and XML (AJAX); • Serverseitige Entwicklung mit Node.js und weiterführende Technologien. 					
<p>Lehrformen: Vorlesung und Übung mit Gruppenarbeiten am Rechner.</p>					
<p>Prüfungsformen: Schriftliche Prüfung von 120 min.</p>					
<p>Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur</p>					

Verwendung des Moduls (in anderen Studiengängen): Masterstudiengang ITS (Wahlfach)
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Markus König Henk Freimuth
Sonstige Informationen: Literatur: 1. jQuery - Das umfassende Handbuch, Rheinwerk Verlag, 2013, ISBN 978-3-8362-2638-7 2. Node.js - Das umfassende Handbuch, Rheinwerk Verlag, 2013, ISBN 978-3-8362-2119-1

Betriebssysteme					
Modul-Nr: 141246	Credits: 5 CP	Workload: 150 h	Semester: 4. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 300
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Grundkenntnisse der Informatik</p>					
<p>Lernziele (learning outcomes): Die Studierenden erlangen ein solides Grundverständnis von modernen Betriebssystemen, ihrer Funktion und ihrer Implementierung. Die Studierenden sind nach Abschluss des Moduls in der Lage, verschiedene Aspekte eines Betriebssystems wie Prozess- und Speicherma- nagement zu verstehen und zu nutzen, sie können dabei verschiedene Designentscheidungen eigenständig analysieren und bewerten. Sie sind in der Lage, bestimmte Aspekte eines Be- triebssystems selbst zu designen und diese argumentativ zu verteidigen.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten • Selbständiges Lernen und Arbeiten • Teamarbeit und Teamfähigkeit <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Algorithmik • IT-Sicherheit <p>Technologien:</p> <ul style="list-style-type: none"> • C 					
<p>Inhalt: Es werden die wichtigsten Grundlagen zu Betriebssystemen vorgestellt. Dazu gehören zum Beispiel:</p> <ul style="list-style-type: none"> • Betriebssystemkonzepte • Prozesse und Threads, Interprozesskommunikation 					

KAPITEL 2. PFLICHTMODULE INFORMATIK

- Scheduling-Mechanismen
- Speicherverwaltung, Speicherabstraktionen, Paging
- Dateisysteme
- Eingabe- und Ausgabeverwaltung
- Algorithmen zur Vermeidung von Deadlocks

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

Lehrformen:

Vorlesung (+Übung)

Prüfungsformen:

Klausur

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. rer. nat. Thorsten Holz

Sonstige Informationen:

Wird das erste Mal im Sommersemester 2015 angeboten.

Literatur:

1. Stallings, W.: Operating Systems: Internals and Design Principles, Pearson Verlag
2. "Moderne Betriebssysteme", Andrew S. Tanenbaum, Pearson 2012.
3. "Modern Operating Systems", Andrew S. Tanenbaum, Pearson 2012.

Datenschutz					
Modul-Nr: 260081	Credits: 4 CP	Workload: 120 h	Semester: 5. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung mit Folien Übungen: Wissens- und Verständnis- abfragen, Anwendung auf Beispiele			Kontaktzeit: 3 SWS	Selbststudium: 75 h	Gruppen- größe: Max 25 Übun- gen Vorlesung 120 Studierende
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Datenschutz befasst sich mit der Frage, wie man Bürger, Arbeitnehmer, Kunden, Patienten etc. vor negativen Auswirkungen durch die Verarbeitung von Daten zu ihrer Person schützen kann. Es besteht die Anforderung an Informatiker, Computersysteme so zu gestalten, dass sie die Umsetzung datenschutzrechtlicher Prinzipien unterstützen. Die Vorlesung befasst sich daher mit den Prinzipien des Datenschutzrechtes und den praktischen Auswirkungen für Informatiker. Dabei wird vor allem Wert darauf gelegt, diese zentralen Prinzipien verstehbar zu machen. Neben dem Datenschutzgrundverordnung werden auch Spezialregelungen behandelt, die z.B. für die Regulierung der Telekommunikation, oder für den Einsatz elektronischer Datenverarbeitung in der Arbeitswelt zum Einsatz kommen. Die DSGVO ist inzwischen auch über den europäischen Raum hinaus ein akzeptierter Standard. Unterschiedliche Rechtsphilosophische Betrachtungen werden thematisiert, um zu vermitteln, wo international Sichtweisen und Fragestellungen divergieren. Insgesamt wird das Thema konstruktiv betrachtet: das Thema Privacy by Design, wird auf allen Ebenen betrachtet. Lernziel der Vorlesung ist es, dass die Studierenden künftig in der Lage sind, zu erkennen, an welchen Stellen ihres beruflichen Wirkens der Datenschutz relevant ist, und wie sie vorgehen müssen, um sich geeignete Informationen oder Sachverstand zu besorgen. Das zu vermittelnde Wissen soll so grundlegend sein, dass man sich auch auf neue Entwicklungen (wie etwa Novellierungen und Ergänzungen des Bundesdatenschutzgesetzes) einstellen kann.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • kennen Studierende die Grundzüge des Datenschutzrechtes, • verstehen Studierende die gesellschaftlichen Hintergründe, • können Datenverarbeitungsprozesse hinsichtlich der Relevanz des Datenschutzrechtes analysieren und • können Lösungsmuster anwenden um Systeme datenschutzfreundlich und datenschutzrechtlichkonform zu gestalten. <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Selbständiges Lernen und Arbeiten 					

KAPITEL 2. PFLICHTMODULE INFORMATIK

- Teamarbeit und Teamfähigkeit
- Präsentation wissenschaftlicher Ergebnisse

Fachspezifische Kompetenzen:

- Rechtliche Kompetenzen

Inhalt:

- Was ist Datenschutz, informationelle Selbstbestimmung und Privacy?
- Welche Folgen haben Verarbeitungen personenbezogener Daten? Woher entstehen diese Folgen?
- Was sind die Prinzipien des Datenschutzes
- Welche Rechte haben die von der Verarbeitung betroffenen Personen?
- Was passiert mit personenbezogenen Daten in vernetzten Systemen?
- Welche organisatorischen und technischen Maßnahmen helfen, personenbezogene Daten zu sichern?
- Was ist Privacy by Design und wie kann das umgesetzt werden?
- Spezielle Bereiche der Datenverarbeitung: Telekommunikation, Wirtschaft, Medizin

Lehrformen:

Vorlesung mit Folien Übungen: Wissens- und Verständnisabfragen, Anwendung auf Beispiele

Prüfungsformen:

Klausur 90min Bonuspunkte für erfolgreiche Übungsteilnahme

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Verwendung des Moduls (in anderen Studiengängen): Wahlmodul in Studiengängen der IT-Sicherheit

Stellenwert der Note für die Endnote: 4 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Dr. Kai-Uwe Loser
Dr. Martin Degeling

Sonstige Informationen:

Literatur:

1. Gola, Peter, Jaspers, Andreas "Das BDSG im Überblick", Datakontext Fachverlag G, 2006

KAPITEL 2. PFLICHTMODULE INFORMATIK

2. Tinnefeld, Marie-Theres, Ehmann, Eugen, Gerling, Rainer W. "Einführung in das Datenschutzrecht. Datenschutz und Informationsfreiheit in europäischer Sicht", Oldenbourg, 2004

Datenbanksysteme					
Modul-Nr: 150304 (Übung: 150305)	Credits: 9 CP	Workload: 270 h	Semester: 5. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Datenbanksysteme (Vorlesung) b) Datenbanksysteme (Übung)			Kontaktzeit: 6 SWS	Selbststudium: 180 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Grundlagen der Informatik und Datenstrukturen.					
Lernziele (learning outcomes): Die Studierenden lernen den in (4+2)-stündigen Lehrveranstaltungen über Datenbanksysteme üblichen Stoff. Die Entwurfstheorie nimmt dabei einen großen Raum ein. Die Benutzung von Datenbanken aus Anwendersicht ist kein Lehrschwerpunkt. Die Studierenden sollen in die Lage versetzt werden, künftig selbst effiziente Datenbanken zu konzipieren, die Anwender nutzen können.					
Inhalt: <ul style="list-style-type: none"> • Strukturen der physikalischer Ebene • Entity-Relationship-Diagramme • Sprachen und Kalküle zur Datendefinition und -manipulation • Optimierung von Abfragen • Deduktionssysteme bei Datenbanken • Strategien der Parallelisierung und der Fehlerbehandlung • Entwurfstheorie: Normalformen, funktionale Anhängigkeiten, verlustlose und abhängigkeitserhaltene Zerlegungen 					
Lehrformen: Vortrag in der Vorlesung, Aufgabenzettel zur Bearbeitung, Aufgabenbesprechung in den Übungen					
Prüfungsformen: 90 minütige Klausur am Ende der Vorlesungszeit, Nachschreibeklausur zu Anfang des Folgesemesters					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					

Stellenwert der Note für die Endnote: 9 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Dr. Edgar Korthauer
Sonstige Informationen: Literatur: 1. A. Kemper, A. Eickler "Datenbanksysteme", Oldenbourg Verlag, 2009 2. R. Elmasri, S. Navathe "Grundlagen von Datenbanksystemen", Pearson, 2009

Pflichtmodule Betriebswirtschaftlehre

Wirtschaftlichkeitsanalyse					
Modul-Nr: 076000 (Übung: 076001)	Credits: 5 CP	Workload: 150 h	Semester: 1. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung mit integrierter Übung			Kontaktzeit: 3 SWS	Selbststudium: 105 h	Gruppen- größe: ca. 150
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls:</p> <ul style="list-style-type: none"> • sollen die Studierenden Grundbegriffe der Wirtschaftlichkeitsanalyse kennen • sollen die Studierenden die verschiedenen Teilgebiete der Wirtschaftlichkeitsanalyse auseinanderhalten können • sollen die Studierenden Aufgaben der Kostenarten-, Kostenstellen- und Kostenträgerrechnung lösen können • sollen die Studierenden Erfolgsgrößen errechnen und beurteilen können • sollen die Studierenden Methoden der Investitionsrechnung sicher anwenden • sollen die Studierenden kompetent mit wirtschaftlichen Fachbegriffen umgehen können • sollen die Studierenden in der Lage sein, sich mit Kaufleuten inhaltlich über die entsprechenden Themen auszutauschen • sollen die Studierenden bei Budgetverantwortung und eigenen Projekten die Grundbegriffe der Wirtschaftlichkeit kennen und berücksichtigen können <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken 					

KAPITEL 3. PFLICHTMODULE BETRIEBSWIRTSCHAFTLEHRE

- Fächerübergreifendes Denken und Arbeiten
- Kritikfähigkeit

Fachspezifische Kompetenzen:

- Rechnungswesen
- Wirtschaftliches Handeln

Inhalt:

Angewandte Informatiker werden sich im Rahmen ihrer Berufstätigkeit als Budgetverantwortliche oder im Rahmen eines Projektmanagements auch regelmäßig mit Frage der Wirtschaftlichkeit ihres Handelns auseinandersetzen haben. Darüber hinaus werden im Kundengespräch und bei der Auftragsabwicklung Kenntnisse von Wirtschaftlichkeitsgrößen und Vorteilhaftigkeitsrechnungen als Verkaufsargumente notwendig sein. Nicht zuletzt wird sich für viele Informatiker sowohl im Rahmen eigener Investitionsüberlegungen als auch bei der Entwicklung von Software-Lösungen für Kunden die Frage nach der Wirtschaftlichkeit von Investitionsalternativen stellen. Die Lehrveranstaltung „Wirtschaftlichkeitsanalyse“ wird die Studierenden mit den Grundlagen der Wirtschaftlichkeitsanalyse vertraut machen und ihnen Methoden vermitteln, mit denen sie derartige Fragestellungen beantworten können.

Gliederung:

1. Anwendung der Wirtschaftlichkeitsanalyse (u.a. Budgetverantwortung u. Projektmanagement, Kundengespräch und Auftragsabwicklung, Investitionsentscheidungen)
2. Grundbegriffe und begriffliche Abgrenzung
3. Die Wirtschaftlichkeit in der Leistungserstellung (u.a. Kostenarten-, Kostenstellen-, Kostenträgerrechnung; Erlöse; Erfolgsermittlung)
4. Verfahren der Wirtschaftlichkeitsrechnung in der Investitionsrechnung (statische und dynamische Verfahren)

Lehrformen:

Vorlesung mit integrierter Übung

Prüfungsformen:

Schriftliche Modulabschlussprüfung von 60 Minuten

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Dr. Barbara Wischermann

Einführung Management Science					
Modul-Nr: 072120 (Übung: 072121/ 072122/ 072123)	Credits: 6 CP	Workload: 180 h	Semester: 2. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppen- größe: ca. 130
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Der vorherige Erwerb betriebswirtschaftlicher Grundkenntnisse und der parallele Besuch des Moduls Projektmanagement werden empfohlen.</p>					
<p>Lernziele (learning outcomes): Dieses Modul bietet eine Einführung in Management Science mit besonderem Schwerpunkt auf mathematischen Optimierungsmethoden. Studierende vertiefen ihre mathematischen und analytischen Fähigkeiten und sind in der Lage, Probleme zu strukturieren und ggf. zu modellieren. Für wichtige Problemstrukturen werden Methoden zur Lösung vermittelt. Studierende erwerben auf Basis der fachinhaltlichen Kenntnisse die Fähigkeit, neue Aufgaben in begrenzter Zeit selbstständig zu lösen.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Selbständiges Lernen und Arbeiten • Projekt- und Zeitmanagement <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Optimierungsmethoden • Mathematik • Wirtschaftliches Handeln 					
<p>Inhalt: In Einführung Management Science werden insbesondere Methoden und Modelle der Unternehmensforschung intensiv behandelt. Diese werden in der Praxis zur Planung und zur Problemlösung in unterschiedlichen Anwendungsbereichen angewandt. Dazu zählen optimierende Methoden z.B. zur Produktionsprogrammplanung in der Fertigung, zur Investitions- und Finanzierungsplanung oder zur Lösung von Transport- bzw. Zuordnungsproblemen in Vertrieb, Logistik und Personalplanung. Die Unternehmensplanung bedient sich ebenfalls häufig quantitativer Methoden. Zusätzlich zur Vorlesung erfolgt in der begleitenden Übung die Vertiefung</p>					

KAPITEL 3. PFLICHTMODULE BETRIEBSWIRTSCHAFTLEHRE

durch Anwendung der Methoden. Die Vorstellung von Standardsoftware zur Lösung größerer Problemstellungen und deren Einsatz ist ebenfalls Bestandteil der Veranstaltung.

Die Modulnote entspricht der Note der Klausur, welche aus Inhalten der Veranstaltungen gebildet wird.

Lehrformen:

Vorlesung (+Übung)

Prüfungsformen:

Klausur 60 Minuten

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Brigitte Werners

Sonstige Informationen:

Das Modul wird über die Fakultät für Wirtschaftswissenschaft angeboten und dort als Teilmodul des Moduls "Quantitative Decision Making" geführt.

Literatur: Veranstaltungsunterlagen und Literaturhinweise werden über Blackboard bereitgestellt.

Nichttechnische Wahlmodule

Nichttechnische Wahlmodule					
Modul-Nr: N/A	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Jedes Semester	Dauer: 1 Semester
Lehrveranstaltungen: frei wählbar; Liste mit empfohlenen Veranstaltungen unter www.ai.rub.de			Kontaktzeit: 3 SWS	Selbststudium: 105 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Abhängig von der gewählten Veranstaltung.					
Lernziele (learning outcomes): Die Teilnehmer erwerben so genannte Schlüsselfähigkeiten in den nichttechnischen Wahlfächern. Vermittelte Kompetenzen: Kernkompetenzen: <ul style="list-style-type: none">• Fächerübergreifendes Denken und Arbeiten					
Inhalt: Die nichttechnischen Wahlfächer erweitern die Soft Skills. Z.B. wird die englische Fachsprache verbessert, in die Grundlagen der Rechtswissenschaften eingeführt oder Grundkenntnisse der Betriebswirtschaft vermittelt. Die Studierenden haben die Möglichkeit eine Auswahl entsprechend der eigenen Interessen zu treffen.					
Lehrformen: frei wählbar; Liste mit empfohlenen Veranstaltungen unter www.ai.rub.de					
Prüfungsformen: Abhängig von der gewählten Veranstaltung.					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					
Stellenwert der Note für die Endnote: 5 / 175					

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Laurenz Wiskott
Dozenten der RUB

Sonstige Informationen:

Unter anderem bieten sich die folgenden Fächer an:

- Englisch für Studierende der Angewandten Informatik
- Andere nichttechnische Veranstaltungen

Aus der Fakultät Wirtschaftswissenschaften dürfen nur die Lehrveranstaltungen belegt werden, die in der Liste (unter www.ai.rub.de) aufgeführt sind. Diese müssen über FlexNow innerhalb der Anmeldephase angemeldet werden.

Praktische Fächer

Intensivkurs C++					
Modul-Nr:	Credits:	Workload:	Semester:	Turnus:	Dauer:
310032	3 CP	90 h	3. Sem.	Wintersemester	1 Semester
Lehrveranstaltungen: Intensivkurs C++			Kontaktzeit: 2-wöchiger Blockkurs Keine	Selbststudium: (ganztägige Präsenzveranstaltung)	Gruppen- größe: 20 - 40 Studierende
Teilnahmevoraussetzungen: Der Kurs richtet sich an Studierende mit Vorkenntnissen in einer imperativen, idealerweise objektorientierten Programmiersprache, z. B. Java oder Python. Dieser Kurs eignet sich nicht für Programmieranfänger.					
Lernziele (learning outcomes): Ziel dieses Kurses ist es, die wesentlichen Paradigmen der Programmiersprache C++ zu erarbeiten und am praktischen Beispiel selbst anzuwenden. Da die Teilnehmer auf einem sehr unterschiedlichen Vorkenntnisstand beginnen, ist es Teil der Praxisaufgaben, noch unbekannte aber für die Aufgaben notwendige Inhalte selbst oder mit Hilfe eines Betreuers zu erschließen. Erfolgreiche Teilnehmer dieses Kurses werden sich in der Lage sehen, kleine bis mittelgroße Softwareprojekte in C++ selbst zu planen, zu konzipieren und zu implementieren, wobei Designentscheidungen getroffen werden, die mit Wiederverwendbarkeit, Lesbarkeit, Effizienz oder Sicherheit begründet werden können.					
Inhalt: <ul style="list-style-type: none"> • die imperative C++-Programmiersprache • Zeigerarithmetik • Klassen • Vererbung • Templates • Die Standard Template Library und Input-Output-Streams • Features des C++14/C++17-Standards 					

KAPITEL 5. PRAKTISCHE FÄCHER

Lehrformen:

Der Blockkurs beginnt täglich mit einer Vorlesung, die in die Thematik des Tages einführt, gefolgt von praktischen Programmierübungen für den Rest des Tages. Diese Aufgaben können allein oder in kleinen Gruppen bearbeitet werden. Zur Kontrolle sollen die Teilnehmer das von ihnen entwickelte Programm vorstellen und Grenzfälle aufzeigen. Zwei Tage des Kurses sind zum Aufholen nicht fertiggestellter Programmieraufgaben vorgesehen. An diesen Tagen werden keine neuen Themen eingeführt.

Prüfungsformen:

Die entwickelten Programme für jeden Tag werden benotet. Der Durchschnitt der Benotungen für jeden Tag ergibt die Gesamtnote für die Prüfung.

Voraussetzungen für die Vergabe von Kreditpunkten:

Neben der erfolgreichen Bearbeitung der Einzeltagesaufgaben als Prüfungsleistung gibt es keine weiteren Voraussetzungen.

Stellenwert der Note für die Endnote: 3 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Daniela Horn, M.A., M.Sc.

Introduction to Python					
Modul-Nr: 310533	Credits: 3 CP	Workload: 90 h	Semester: 3-6. Sem.	Turnus: Sommersemester	Dauer: 2 weeks (block)
Lehrveranstaltungen: Programmierpraktikum: „Introduction to Python“			Kontaktzeit: 3.2	Selbststudium: 42 h	Gruppen- größe: 10-36
<p>Teilnahmevoraussetzungen: Vorlesung "Informatik 1" und "Programmieren in C"</p> <p>Empfohlene Vorkenntnisse: We expect fluency in one other programming language and familiarity with concepts like - loops and control structures (while, for, if ...), - basic data types and structures (boolean, int, float, string, arrays, ...), - functions, and - object-oriented programming. These concepts will not be taught separately. A solid understanding of basic maths and algorithms is also recommended for a successful project.</p>					
<p>Lernziele (learning outcomes): After the successful completion of this course the students</p> <ul style="list-style-type: none"> • will know and be able to apply basic syntax and structure of Python 3, • will understand numerical representations and processing of data using Numpy, • will have gained first practical experience in planning and conducting a small project in a team using Python 3. 					
<p>Inhalt: Python is a programming language that is wide-spread among scientists due to its readability and powerful standard libraries. This practical course teaches Python 3 to students with prior experience in other programming languages. In addition to introducing the language itself, we will focus on scientific computing including vectors and matrices as well as data processing and possibly simple machine learning. All course-work is done in teams of two. During the first week, participants will work on Jupyter notebooks autonomously and discover Python 3 in a largely self-taught manner. Teaching assistants are present and support is provided if required. During the second week, participants will implement a project in Python 3 using the previously acquired skills. We provide a default project, usually from the area of machine learning. Alternatively, own project ideas can be realized if discussed early on with the Teaching Assistants.</p>					
<p>Lehrformen: Two week lab-course (Blockkurs in der vorlesungsfreien Zeit)</p>					
<p>Prüfungsformen: Team-project and -presentation with additional individual questions. Requirement for project/presentation is successful completion of the notebook exercises.</p>					

Voraussetzungen für die Vergabe von Kreditpunkten:

Successfully demonstrating command of Python/numpy in project presentation and questions.

Verwendung des Moduls (in anderen Studiengängen): The lab course is open for students of other faculties, if they fulfill the requirements. They will be graded similar to students from Applied Computer Science. If there are more applicants than seats, Applied Computer Science students will be treated with priority. After that, the seats will be allocated on a first-come, first-serve basis based on enrollment emails (see below).

Stellenwert der Note für die Endnote: 3 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Merlin Schüler, Prof. Dr. rer. nat. Laurenz Wiskott

Sonstige Informationen:

The course language is English.

In addition to registering via your examination office/FlexNow/..., please send an eMail to python@ini.rub.de Include your name, your student ID, and your study program.

Programmierpraktikum Android					
Modul-Nr: 260086	Credits: 3 CP	Workload: 90 h	Semester: 3-4. Sem.	Turnus: Sommersemester (nicht im SS21!)	Dauer: 1 Semester
Lehrveranstaltungen: Block-Praktikum			Kontaktzeit: 2 SWS	Selbststudium: 60 h	Gruppen- größe: 14-20
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Das Praktikum richtet sich an Studierende mit Vorkenntnissen in Java (oder einer vergleichbaren Objekt-orientierten Sprache) und soll den Einstieg in die App-Entwicklung für Android ermöglichen. Für die Entwicklung sollen die Teilnehmer auf eigene Hardware (PC und bestenfalls Android Smartphone) zurückgreifen. Die Studierenden lernen nach kurzer Einführung in das Themengebiet „Android-App-Entwicklung“ selbstständig Lösungen für häufige Probleme zu entwickeln. Dabei wird das Nutzen komplexer Dokumentation erlernt, sowie die Recherche bei spezifischen Problemen sowie die Nutzung von Entwicklerwerkzeugen. Neben den spezifischen Inhalten, werden Themen wie Fehlerbehandlung, Internationalisierung, Modularisierung, Usability/User-Expierience und Barrierefreiheit vertieft.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Literaturrecherche und Dokumentation • Projekt- und Zeitmanagement • Selbständiges Lernen und Arbeiten • Teamarbeit und Teamfähigkeit <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Software-Entwurfsmethoden • Programmieren <p>Technologien:</p> <ul style="list-style-type: none"> • Python • Android Studio 					
<p>Inhalt: Auszug Themen: „Android Studio“ IDE, Views, Activities, Intents, Gestures, RESTful Webservices, NFC.</p>					

KAPITEL 5. PRAKTISCHE FÄCHER

Lehrformen:

Das Praktikum wird als einwöchige Blockveranstaltung in der vorlesungsfreien Zeit durchgeführt. Nach kurzen Einführungsvorträgen durch die DozentInnen arbeiten die Studierenden selbstständig zusammenhängende Praktikumsaufgaben ab. Jede TeilnehmerIn erarbeitet eine eigenständige Lösung, der Austausch mit anderen TeilnehmerInnen und gegenseitige Hilfe sind dabei gewünscht. Nach jeder Teilaufgabe erhalten die TeilnehmerInnen Feedback zu ihrer Lösung von den DozentInnen.

Prüfungsformen:

Die Praktikumsaufgaben werden durch die DozentInnen bewertet.

Voraussetzungen für die Vergabe von Kreditpunkten:

Regelmäßige Teilnahme am Praktikum und Bearbeitung der Aufgaben.

Stellenwert der Note für die Endnote: 3 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Thomas Herrmann Christopher Lentzsch M.Sc.

Grundlagen Roboterprogrammierung					
Modul-Nr: 138970	Credits: 3 CP	Workload: 90 h	Semester: 4. Sem.	Turnus: Sommersemester	Dauer: 1 Woche
Lehrveranstaltungen: a) Praktikum			Kontaktzeit: 24h keine	Selbststudium: 66 h	Gruppen- größe: 6 Studierende
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Teilnahme an der Vorlesung "Grundlagen der Automatisierungstechnik"</p>					
<p>Lernziele (learning outcomes): Das Praktikum "Grundlagen der Roboterprogrammierung" soll den AI-Studierenden die Möglichkeit bieten, in einem komprimierten Kurs die Roboterprogrammierung an Industrierobotern zu erlernen. Ziel des einwöchigen Kurses ist die selbständige Entwicklung und Umsetzung von einfachen Roboterbewegungsprogrammen mit Peripherie-Interaktion, bspw. zur Greifer- und Sensorik-Integration.</p> <p>Mit den erlernten Fähigkeiten werden die Studierenden in die Lage versetzt, Industrieroboter zu programmieren sowie industrielle Aufgabenstellungen in der robotergestützten Automatisierung an einem realen System umzusetzen und es wird ein geeignetes Wissen für die mögliche spätere wissenschaftliche Vertiefung sowie die Berufstätigkeit im Automatisierungs- und Produktionssystemsbereich erlangt. Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • sind die Studierenden in der Lage Industrieroboter so zu programmieren, dass einfache Peripherie-Interaktionen möglich sind • können die Studierenden selbstständig Aufgabenstellungen entwickeln, die durch geeignete Roboterprogrammierung gelöst werden können • können die Studierenden Ihre abgeschlossenen Projekte in einer Präsentation unter Verwendung des entsprechenden Fachvokabulars vorstellen 					
<p>Inhalt: Diese Inhalte werden den Studierenden schrittweise und angeleitet näher gebracht und in selbstständiger Arbeit vertieft: Angefangen von der Arbeitssicherheit beim Umgang mit Industrierobotern sowie dem Aufbau und der Funktionsweise eines Robotersystems, über das Erstellen von einfachen Bewegungsprogrammen und dem Einmessen von Werkzeugen und Koordinatensystemen, endet der Kurs mit der Realisierung einer Handhabungsaufgabe, in der die Studierenden ein vollständiges Roboterprogramm inklusive Greiferansteuerung am Robotersystem umsetzen. Zu Beginn des Kurses werden die einfacheren Programme direkt per Teach-In am Roboter programmiert und ausgetestet. Mit der Steigerung der Komplexität der Roboterprogramme erfolgt die Programmierung dann in der Robotersprache RAPID, jeweils mit abschließenden Überprüfungen am realen System.</p> <p>Im Bereich der Kompetenzen im Kontext Digitalisierung ist der Simulationsteil angesiedelt, dieser beinhaltet die Einführung in ein kommerziell verfügbares und industriell einge-</p>					

KAPITEL 5. PRAKTISCHE FÄCHER

setztes 3D-Robotersimulations- und Offline-Programmiersystem.
Lehrformen: Praktikum in 6er-Gruppe
Prüfungsformen: Präsentation der programmierten Aufgaben
Voraussetzungen für die Vergabe von Kreditpunkten: Erfolgreiche Beteiligung an der Gruppenarbeit und der -abschlusspräsentation
Stellenwert der Note für die Endnote: 3 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Dr. Alfred Hypki

KAPITEL 5. PRAKTISCHE FÄCHER

Introduction to CL with Python					
Modul-Nr: 001	Credits: 3 CP	Workload: 90 h	Semester: 4. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Praktikum			Kontaktzeit: MISSING	Selbststudium: 2 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine					
Lernziele (learning outcomes): This class introduces you to linguistic models for syntactic parsing involving the concepts of constituency and dependency. You will learn to parse data using NLP libraries like NLTK and spaCy in Python.					
Inhalt: This course introduces you to a computational linguistist's perspective on language, data, and programming. You will learn to write programs solving first small problems in Python from scratch.					
Lehrformen: Vorlesung und Übung					
Prüfungsformen: Übungsaufgaben					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestehen der wöchentlichen Übungsaufgaben (Einzel- und Gruppen-Abgabe von Programmieraufgaben, Erstellen kurzer Textantworten auf Fragen oder Teilnahme an Diskussionen im Forum)					
Stellenwert der Note für die Endnote: 3 / 175					
Modulbeauftragte/r und hauptamtlich Lehrende: Mirjam Koch, M.Sc.					

Studienprojekt					
Modul-Nr: N/A	Credits: 8 CP	Workload: 240 h	Semester: 3.-6. Sem.	Turnus: Jedes Semester	Dauer: 1 Semester
Lehrveranstaltungen: Praktikum/Projektarbeit			Kontaktzeit: 2 SWS	Selbststudium: 210 h	Gruppen- größe: 2-6
<p>Teilnahmevoraussetzungen: 90 LP, siehe Prüfungsordnung</p> <p>Empfohlene Vorkenntnisse: Bitte entnehmen Sie die projektspezifischen Vorkenntnisse dem jeweiligen Studienprojekt.</p>					
<p>Lernziele (learning outcomes): Ziele des Studienprojekts sind:</p> <ul style="list-style-type: none"> • Die Anwendung des erlernten Fachwissens. • Der Erwerb zusätzlicher Fachkompetenz gemäß der jeweiligen projektspezifischen Aufgabenstellung. • Die Schulung hinsichtlich der Erarbeitung eigener Lösungsstrategien. • Die Schulung hinsichtlich Arbeitsteilung und Zusammenarbeit im Team (Teamfähigkeit und Projektorganisation). • Erwerb weiterer Kompetenz im Hinblick auf die Dokumentation und die Präsentation von Ergebnissen. <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Kritikfähigkeit • Projekt- und Zeitmanagement • Präsentation wissenschaftlicher Ergebnisse • Teamarbeit und Teamfähigkeit • Fächerübergreifendes Denken und Arbeiten 					
<p>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, die Sie unter http://www.ai.rub.de/studierende/studienprojekte-und-abschlussarbeiten/studienprojekt.html finden. In begründeten Fällen (Industriekooperation etc.) können Studienprojekte auch au-</p>					

KAPITEL 5. PRAKTISCHE FÄCHER

ßerhalb 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:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 8 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Thomas Herrmann

Dozenten der RUB

Projektmanagement					
Modul-Nr: 260082	Credits: 2 CP	Workload: 60 h	Semester: 4-5. Sem.	Turnus: Winter- und Sommersemes- ter	Dauer: 1 Semester
Lehrveranstaltungen: a) Inverted Classroom Veranstaltung			Kontaktzeit: 2 SWS	Selbststudium: 30 h	Gruppen- größe: 10-25
Teilnahmevoraussetzungen: Teilnahme an einem Studienprojekt					
<p>Lernziele (learning outcomes): Studierende erwerben Grundkenntnisse und Fähigkeiten zum IT-Projektmanagement. Sie lernen Softwareentwicklungsprozesse, Projektabläufe und -phasen kennen sowie deren Zusammenhänge. Die theoretischen Erkenntnisse werden in Gruppen anhand ausgewählter Methoden und Instrumente des Projektmanagements an ihrem konkreten Studienprojekt vertieft.</p> <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Koordinationsfähigkeit und Aufgabenaufteilung • Projekt- und Zeitmanagement • Kritikfähigkeit • Präsentation von Projektergebnissen • Teamarbeit und Teamfähigkeit • Rhetorik und sprachliche Kompetenz 					
<p>Inhalt: Die Studierenden beschäftigen sich mit Methoden und Werkzeugen des Projektmanagements in klassischen und agilen Projektszenarien. Dabei werden alle Projektphasen von der Initialisierung über das Controlling bis hin zum Projektabschluss behandelt. Im Fokus stehen dabei die parallel von den Studierenden bearbeiteten Bachelor-Studienprojekte und die konkrete Anwendung von Projektmanagement auf ihre eigene Arbeit. Verstärkt durch den Inverted-Classroom-Ansatz ist es den Teams möglich in eigenem Lern-tempo die bereitgestellten Unterlagen und Methoden zu erlernen und in den über das Semester verteilten Präsenzterminen können dann Fragen und aktuelle Probleme besprochen und vertieft werden.</p>					
<p>Lehrformen: Interaktive Lernfolien zum Selbstlernen. Präsenzveranstaltung zur Vertiefung der Themen und praktische Anwendung der Methoden auf die eigenen Studienprojekte. Besprechung und Austausch über aktuelle Probleme oder Hindernisse beim Management des Studienprojektes.</p>					

KAPITEL 5. PRAKTISCHE FÄCHER

<p>Prüfungsformen: Zwei schriftliche Hausaufgaben im Laufe des Semesters (jeweils ca. 2-5 Seiten in Gruppenarbeit). Ein Abschlussvortrag (ca. 15 Minuten).</p>
<p>Voraussetzungen für die Vergabe von Kreditpunkten: Abgabe von zwei Hausaufgaben und aktive Teilnahme am Abschlussvortrag der Gruppe.</p>
<p>Stellenwert der Note für die Endnote: 2 / 175</p>
<p>Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Thomas Herrmann Marcel Schmittchen Msc.</p>
<p>Sonstige Informationen: Literatur:</p> <ol style="list-style-type: none">1. Schelle, H.; Ottmann, R.; Pfeiffer, A.: ProjektManager. GPM Deutsche Gesellschaft für Projektmanagement e.V. (Hrsg.), Nürnberg, 2. Auflage 2005.2. Tiemeyer, E. (Hrsg.): Handbuch IT-Projektmanagement. Carl Hanser Verlag, München 2010.3. Mangold, P.: IT-Projektmanagement kompakt. Spektrum Akademischer Verlag, München 2004.

Bachelorarbeit

Bachelorarbeit					
Modul-Nr: N/A	Credits: 12 CP	Workload: 360 h	Semester: 3.-6. Sem.	Turnus: Jedes Semester	Dauer: 1 Semester
Lehrveranstaltungen: Praktikum/Projektarbeit			Kontaktzeit: 0 SWS	Selbststudium: 360 h	Gruppen- größe: beliebig
<p>Teilnahmevoraussetzungen: Die Ausgabe des Themas der Bachelorarbeit kann erst dann erfolgen, wenn der Studierende mindestens 120 Leistungspunkte erreicht hat. Idealerweise sollte die Bachelor-Arbeit jedoch den Abschluss des Bachelorstudiums darstellen.</p> <p>Empfohlene Vorkenntnisse: Fachspezifisch von der Themenstellung abhängig.</p>					
<p>Lernziele (learning outcomes): Nach erfolgreichem Abschluss des Moduls</p> <ul style="list-style-type: none"> • können Studierende selbstständig und fristgerecht ein wissenschaftliches Thema bearbeiten von der Recherche bis zur Dokumentation der Resultate • können Studierende geeignete wissenschaftliche Verfahren und Methoden, die sie im Studium kennengelernt haben, auswählen und anwenden, um ein konkretes Problem zu lösen • können Studierende ihre Ergebnisse kritisch mit dem Stand der Forschung vergleichen und evaluieren • können Studierende ihre eigenen Ergebnisse angemessen in Wort und Schrift darstellen <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Kritikfähigkeit • Projekt- und Zeitmanagement • Präsentation wissenschaftlicher Ergebnisse 					

KAPITEL 6. BACHELORARBEIT

- Rhetorik und sprachliche Kompetenz
- Fächerübergreifendes Denken und Arbeiten
- Wissenschaftliches Arbeiten und Schreiben

Inhalt:

Die Bachelorarbeit ist eine schriftliche Prüfungsarbeit. Es wird die selbstständige Bearbeitung einer anspruchsvollen Aufgabe der Angewandten Informatik unter Anwendung der im Bachelorstudium erworbenen Kenntnisse und Methoden erwartet. Im Anschluss an die Bearbeitung der Bachelorarbeit werden die Ergebnisse in Form eines Kolloquiumsvortrags präsentiert.

Lehrformen:

Praktikum/Projektarbeit

Prüfungsformen:

Abschlussarbeit

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 12 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Studiendekan 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 Bachelorarbeit schreiben möchte. Die meisten Lehrstühle veröffentlichen ihre angebotenen Themen. Oft werden Themen aber auch erst nach Absprache mit dem Studierenden gestellt, wobei der Studierende ein Vorschlagsrecht hat. Die Anmeldung für die Bachelorarbeit erfolgt beim Prüfungsamt der Angewandten Informatik. Für die Anmeldung ist das persönliche Erscheinen sowie die Vorlage des aktuellen Studentenausweises bzw. der Immatrikulationsbescheinigung notwendig.

Ingenieurinformatik

Digitaltechnik					
Modul-Nr: 141300	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 350
<p>Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse:</p> <ul style="list-style-type: none"> • Grundlagen der Elektronik <p>Erforderlich sind zudem elementare Kenntnisse in:</p> <ul style="list-style-type: none"> • Grundlagen der Elektrotechnik • Mathematik 					
<p>Lernziele (learning outcomes): Die Studierenden erwerben umfassende Kenntnisse in den Bereichen Boolesche Algebra, Struktur und Funktionalität digitaler Grundschaltungen, Kostenoptimierung digitaler Funktionsgruppen, Techniken zur taktysynchronen Verarbeitung von Daten, Kodierung und Verarbeitung von Daten, Struktur und Funktionalität solcher Grundfunktionalitäten, die insbesondere zentrale Bestandteile in Mikroprozessorarchitekturen und deren Umgebung sind. Richtlinien für den Wissenstransfer sind die schaltungstechnischen Möglichkeiten und Grenzen moderner CMOS-Logikstrukturen, um den Studierenden gleichzeitig auch aktuelle Entwicklungstrends in einer sich rasant entwickelnden digitalen Anwendungswelt besser verständlich zu machen.</p> <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Fächerübergreifendes Denken und Arbeiten <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Logik und Schaltungen 					

Inhalt:

Vorlesungsinhalt:

- Historischer Rückblick, Motivation Digitaltechnik
- Boolesche Algebra
- Gatterlaufzeiten, Timing-Analyse, kritischer Pfad
- Zahlensysteme, Zahlenkodierungen, Fehlererkennung und Korrektur, Fest- und Fließkommadarstellungen
- Rechenschaltungen, arithmetisch logische Einheit (ALU),
- Flankendetektoren, bi-, mono- und astabile Schaltungen, transparente und nicht-transparente Flip-Flops (FF)
- Frequenzteiler, Zähler (asynchron, synchron), Automaten, Schieberegister
- Speicher: S-RAM, D-RAM, ROM, ... (Aufbau und Organisationsformen)
- taktsynchrone Techniken zur Datenverarbeitung
- ALU in Umgebungen zur Mikroprogrammierung, Mikroprogrammierung
- Konzepte zur serielle Datenübertragung
- Grundlagenidee von A/D- und D/A-Wandlern
- Konzept: skalierbare Standard-Logik-Zellen, CMOS-Logik
- Übersicht: Logikanalyse, Tools zur Logikanalyse, HDL Entwurfssprachen
- Moore, More than Moore

Lehrformen:

Vorlesung (+Übung)

Prüfungsformen:

Klausur, 120 Minuten

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Jürgen Oehm

Sonstige Informationen:

Literatur:

1. Katz, Randy H. "Contemporary Logic Design", Prentice Hall, 1993

KAPITEL 7. INGENIEURINFORMATIK

2. Seifart, Manfred, Beikirch, Helmut "Digitale Schaltungen", Verlag Technik, 1998
3. Borucki, Lorenz, Stockfisch, Georg "Digitaltechnik", Teubner Verlag, 1989
4. Pernards, Peter "Digitaltechnik I. Grundlagen, Entwurf, Schaltungen", Hüthig, 2001
5. Fricke, Klaus "Digitaltechnik. Lehr- und Übungsbuch für Elektrotechniker und Informatiker", Vieweg, 2005
6. Lipp, Hans Martin, Becker, Jürgen "Grundlagen der Digitaltechnik", Oldenbourg, 2005
7. Tietze, Ulrich, Schenk, Christoph, Gamm, Eberhard "Halbleiter - Schaltungstechnik", Springer, 2002
8. "Handbuch der Elektronik. Digitaltechnik", Medien Institut Bremen, 1999
9. Weste, Neil H. E., Eshragian, Karman, Eshragian, Kamran "Principles of CMOS VLSI Design: A Systems Perspective", Addison Wesley Longman Publishing Co, 1993
10. Wuttke, Heinz-Dieter, Henke, Karsten "Schaltssysteme. Eine automatenorientierte Einführung", Pearson Studium, 2002
11. Siemers, Christian, Sikora, Axel "Taschenbuch Digitaltechnik", Hanser Fachbuchverlag, 2002
12. Schiffmann, Wolfram, Schmitz, Robert "Technische Informatik 1. Grundlagen der digitalen Elektronik", Springer, 2003

Geometrische Modellierung und Visualisierung					
Modul-Nr: 129008	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Objektorientierte Modellierung					
<p>Lernziele (learning outcomes): Im Rahmen des Moduls werden die wesentlichen Grundlagen der geometrischen Modellierung sowie elementare Techniken der Visualisierung erlernt. Mit Abschluss des Moduls befinden Sie sich in der Lage, geometrische und visualisierungsbezogene Aufgaben aus Forschung und Praxis unter Verwendung aktueller Methoden der Mathematik und Information zielgerecht zu lösen.</p> <p>Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • kennen Studierende Techniken zur digitalen Visualisierung von Geometrie, • sind Studierende darin geschult räumlich zu denken, • verstehen Studierende wie Geometrie transformiert werden kann, • haben Studierende ein Verständnis über Programmierung erlernt, • wissen Studierende wie Geometrie persistent gespeichert werden kann, • sind Studierende darauf vorbereitet erlerntes auf international genutzte Anwendungen mit visueller Verarbeitung von Geometrie zu übertragen 					
<p>Inhalt:</p> <ul style="list-style-type: none"> • Geometrische Modelle • Affine Abbildungen und Differentialgeometrie • Freiformkurven und -flächen • Boundary Repräsentationen • Constructive Solid Geometry • Quadrees, Octrees • Zerlegung und Triangulierung • Visualisierungstechniken 					

KAPITEL 7. INGENIEURINFORMATIK

Lehrformen: Vorlesungen, Übungen am Computer
Prüfungsformen: Klausur, Dauer: 180 min
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Verwendung des Moduls (in anderen Studiengängen): Angewandte Informatik (Bachelor); Bauingenieurwesen (Master)
Stellenwert der Note für die Endnote: 6 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Markus König Andre Vonthron, M.Sc. Marcel Stepien, M. Sc.
Sonstige Informationen: Literatur: 1. Gerald Farin: Curves and Surfaces for CAGD - A Practical Guide. 5th Edition. San Francisco: Morgan Kaufmann. 2002 2. David Salomon: Curves and Surfaces for Computer Graphics. New York: Springer. 2006 3. Raymond Zavodnik, Herbert Kopp: Graphische Datenverarbeitung - Grundzüge und Anwendungen. München u.a.: Carl Hanser

Grundlagen der Automatisierungstechnik					
Modul-Nr: 135110	Credits: 6 CP	Workload: 180 h	Semester: 3-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Grundlagen der Automatisierungstechnik			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppengröße: beliebig
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Zielsetzung:</p> <ul style="list-style-type: none"> • Die Studierenden sollen in der Lage sein, aktuelle Entwicklungen und Trends in der Automatisierungstechnik darzulegen sowie Entwicklungsprozesse für automatisierte technische Systeme erläutern und die entsprechenden Entwicklungsmethoden anwenden zu können. • Sie sollen durch Absolvieren des Kurses in die Lage gebracht werden, das Funktionsprinzip und den Hardware-Aufbau einer SPS darzulegen und Automatisierungsaufgaben im Bereich der SPS- und NC-Programmierung mit methodischer Vorgehensweise zu bearbeiten. • Zudem sollen sie die Kenntnisse erlangen, Robotersysteme für den Einsatz in unterschiedlichen Automatisierungsaufgaben kritisch zu bewerten, geeignete Systeme auszuwählen sowie Sicherheitsrisiken der Automatisierungstechnik zu beurteilen. <p>Kenntnisse:</p> <ul style="list-style-type: none"> • Die Studierenden kennen wesentliche Methoden und Verfahren der Ingenieurwissenschaften / des Maschinenbaus, verfügen über entsprechendes Fachvokabular und kennen Anwendungsbeispiele. • Die Studierenden kennen vertiefte ingenieurwissenschaftliche Grundlagen im Bereich ihres Studienschwerpunkts. <p>Fertigkeiten:</p> <ul style="list-style-type: none"> • Die Studierenden verfügen über die Fähigkeit zu vernetztem und kritischem Denken. • Die Studierenden praktizieren erste Ansätze wissenschaftlichen Lernens und Denkens. • Die Studierenden können ingenieurtechnische Probleme modellieren und lösen. • Die Studierenden können komplexe mathematische Problemstellungen in physikalischen Systemen (ggf. fachübergreifend) mit geeigneten Methoden lösen. 					

KAPITEL 7. INGENIEURINFORMATIK

- Die Studierenden haben die Fähigkeit zu vernetztem und kritischem Denken ausgebaut und sind in der Lage etablierte Methoden und Verfahren auszuwählen und anzuwenden.

Kompetenzen:

- Die Studierenden verfügen über fachübergreifende Methodenkompetenz.
- Die Studierenden können Erkenntnisse / Fertigkeiten auf konkrete maschinenbauliche / ingenieurwissenschaftliche Problemstellungen übertragen.

Inhalt:

Nach einem allgemeinen historischen Überblick über die Entwicklung der Automatisierungstechnik werden wesentliche Entwicklungsmethoden und Notationen für Automatisierungsaufgaben vorgestellt. Ein Schwerpunkt der Lehrveranstaltung steht die SPS mit ihrem Hardwareaufbau und dem Echtzeitbetriebssystem. Die SPS-Programmierung wird in Laborübungen vertieft. Dabei spielt die Signalverarbeitung von der Erfassung der Sensorsignale über die Verarbeitung im Steuerungsalgorithmus bis zur Ausgabe der Steuerbefehle an die Stellglieder eine wesentliche Rolle. Die Anwendung des PC für die industrielle Automatisierung und die dezentrale Signalerfassung und -ausgabe werden exemplarisch behandelt. Einen weiteren Schwerpunkt bilden die vermittelten Grundlagen zur industriellen Robotik. Dabei wird zunächst die Entwicklung der Industrierobotik dargelegt. Des Weiteren werden die wesentlichen Bestandteile eines Robotersystems gelehrt und verschiedene Industrierobotertypen und deren Einsatzgebiete in der Automatisierungstechnik vorgestellt. Die prinzipielle Funktionsweise von sowohl Robotersteuerungen als auch numerischen Steuerungen werden in Vorlesungs- und Übungseinheiten vertieft. Die Lehrveranstaltung schließt mit einer Einführung in die Grundlagen der Kommunikationstechnik, Sensorik und Sicherheitstechnik im Themenfeld der Automatisierung ab. Vorträge von Gastreferenten aus Industrie und Forschung zeigen praxisnahe Anwendungsbeispiele aus der Automatisierungstechnik auf und ergänzen somit die Lehrveranstaltung. Übungen dienen der weiteren Vertiefung des gelesenen Lehrstoffes.

Lehrformen:

Vorlesungseinheiten und Übungseinheiten zur Vertiefung der Vorlesungsinhalte; In den Vorlesungseinheiten werden ebenfalls kleinere Übungsaufgaben (zum Teil in Gruppenarbeiten) durchgeführt.

Prüfungsformen:

Schriftliche Prüfung (180 Minuten) in der Prüfungszeit nach Vorlesungsende

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Verwendung des Moduls (in anderen Studiengängen): Studiengang Maschinenbau, Studiengang SEPM, Studiengang Angewandte Informatik, Studiengang Elektrotechnik

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr.-Ing. Bernd Kuhlenkötter

Vorlesungsbetreuer: Thomas Glaser, M. Sc.; Paul Glogowski, M. Sc. Übungsbetreuer: Dennis

Möllensiep, M. Sc.; Michael Krampe, M. Sc.; Mathias Weißköppel, B. Sc.; Stefanie Spies, M. Sc.;

Sonstige Informationen:

Literatur:

1. Abel, D.: Petri-Netze für Ingenieure - Modellbildung und Analyse diskret gesteuerter Systeme, Springer-Verlag, Berlin, 1990.
2. John, K.-H., Tiegelkamp, M.: SPS-Programmierung mit IEC 1131-3, Springer Verlag, Berlin, 2009.
3. Stein, G.: Automatisierungstechnik in der Maschinentechnik, Hanser Verlag, München, 1994.
4. Weck, M.: Werkzeugmaschinen Fertigungssysteme, VDI Verlag, 2006.
5. Wellenreuther, G.; Zastrow, D.: Steuerungstechnik mit SPS
6. Vieweg Verlag, Wiesbaden, 1998.
7. Wloka, Dieter W.: Robotersysteme Band 1: Technische Grundlagen
8. Springer Verlag, Berlin, 1992.

DIN-Normen und Richtlinien

Mensch-Maschine-Interaktion					
Modul-Nr: 260083	Credits: 5 CP	Workload: 150 h	Semester: 3-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung und Übung integriert			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: 15-25
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Die Studierenden lernen, was bei der Gestaltung von Benutzungsschnittstellen heute und auch bei zukünftigen Formen der Mensch-Maschine-Interaktion zu berücksichtigen ist, und wie man beim Entwicklungsprozess vorgeht.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Fächerübergreifendes Denken und Arbeiten <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Entwicklungsprozesse 					
<p>Inhalt:</p> <ol style="list-style-type: none"> 1. Relevante Eigenschaften des Menschen und seines Arbeitskontextes - Physiologie - Wahrnehmungspsychologie - Kognitionspsychologie - Arbeitspsychologie 2. Usability-Engineering - Methoden der ergonomischen Gestaltung der Mensch-Computer-Schnittstelle <ul style="list-style-type: none"> • Methoden der Anforderungsanalyse • Maskengestaltung • Prototyping • Usability-Testing 3. 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"> • Arbeitswissenschaft und Ergonomie 					

KAPITEL 7. INGENIEURINFORMATIK

- Psychologie
- Organisation und Management
- Rechtliche Aspekte
- Betriebswirtschaftliche Aspekte
- Kommunikationstheorie
- Datenschutz

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

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

Lehrformen:

Abwechslung von Vorlesungsinhalten, Seminarbeiträgen der Studierenden und Präsenzübungen

Prüfungsformen:

Semesterbegleitend; Seminarbeitrag und Klausur

Voraussetzungen für die Vergabe von Kreditpunkten:

Seminarbeitrag (50% der Endnote) und Klausur (60 Minuten, 50% der Endnote)

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Thomas Herrmann

Markus Jelonek Msc.

Sonstige Informationen:

Nicht im WS 21/22!

Im Wintersemester 2019/20 kann die Vorlesung "Mensch-Maschine Interaktion" nicht von Prof. Herrmann angeboten werden. Er hat mit Prof. Schneegaß an der Universität Duisburg-Essen (Campus Essen) vereinbart, dass die dort durchgeführte Veranstaltung Human-Computer Interaction ersatzweise von maximal 14 Studierenden wahrgenommen werden kann:

<https://www.hci.wiwi.uni-due.de/aktuelles/einzelansicht/vorlesungen-ws-20192019893/>

Die Anmeldung kann bis zum 20. September 2019 erfolgen. Die Reihenfolge der Anmeldung ist ausschlaggebend. Melden Sie sich bitte beim Lehrstuhl IMTM über marina.konrad@ruhr-

uni-bochum.de an. Zusätzlich ist die Anmeldung zur Prüfung über Flexnow innerhalb der Anmeldephase vorzunehmen.

Literatur:

1. Bernhard Preim, R.D. (2010). Interaktive Systeme. Springer-Verlag Berlin-Heidelberg.
2. Herczeg, M. (2005). Software-Ergonomie. Oldenbourg Wissenschaftsverlag.

Virtuelle Produktmodellierung und Visualisierung					
Modul-Nr: 135060	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Virtuelle Produktmodellierung und Visualisierung			Kontaktzeit: V: 2 SWS; Ü: 2	Selbststudium: 120 h	Gruppengröße: beliebig
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • verfügen Studierende über ein breites, integriertes Wissen über die Herausforderungen moderner Produktentstehungsprozesse und die resultierenden Anforderungen an Softwaresysteme zur Virtuellen Produktmodellierung und -visualisierung. • kennen und verstehen Studierende wesentliche Methoden und Verfahren der Virtuellen Produktmodellierung und -visualisierung einschließlich der angrenzenden Gebiete und der wissenschaftlichen informationstechnischen Grundlagen. Indem sie praktische Beispiele und Aufgaben mit entsprechender Anwendungssoftware bearbeiten, können sie die erlernten Fertigkeiten im Umgang mit Softwaresystemen auf konkrete konstruktionsstechnische Problemstellungen übertragen, um diese modellieren und lösen zu können. • haben Studierende ein umfassendes Verständnis vom Zusammenwirken der Softwaresysteme und Produktdatenmodelle innerhalb der verschiedenen Prozessketten in der Produktentstehung und können kritisch die Eignung von Methoden zur Virtuellen Produktmodellierung und visualisierung zur Konzeption, Konstruktion, Optimierung, Darstellung, Fertigungsvorbereitung und Dokumentation von Produkten differenzieren und beurteilen. • können Studierende Aufgabenstellungen der Virtuellen Produktmodellierung und -visualisierung reflektieren und bewerten sowie selbstgesteuert verfolgen • können Studierende kooperativ Aufgabenstellungen der Virtuellen Produktmodellierung und -visualisierung in heterogenen Gruppen bearbeiten, Abläufe und Ergebnisse begründen sowie über Sachverhalte umfassend kommunizieren 					
<p>Inhalt: Die Veranstaltung vermittelt Methoden und Werkzeuge zur "Virtuellen Produktmodellierung und -visualisierung", insbesondere das dazu erforderliche Grundlagenwissen und die relevanten methodischen Aspekte der systematischen Produktentwicklung. Schwerpunkte bilden dabei die verschiedenen CAD-Modellierungsmethoden (z.B. 3D Flächen und Volumenmodellierung, parametrische Modellierung, Baugruppenmodellierung) entsprechend der Anforderungen aus der Konstruktionsaufgabe sowie die Kombination von Verfahren zur durchgängigen Abbildung von Prozessketten (z.B. für Digital Mockup (DMU), Virtuelle und Augmentierte Realität (VR/AR), Auslegungs- und Nachweisberechnungen, Analyse und Simulation, Addi-</p>					

KAPITEL 7. INGENIEURINFORMATIK

tive Manufacturing, Produktion (CAM), Digital Factory, Styling, Elektro/Elektronik-CAD) im Produktlebenszyklus mit Aspekten der Integration von Modellen und Werkzeugen.
Lehrformen: Vorlesungen, Übungen, Gruppenarbeiten, Moodle Unterstützung
Prüfungsformen: <ul style="list-style-type: none">• Schriftliche Klausur mit Fragen und Aufgaben zum Stoff der Vorlesung und der Übung• Dauer: 90 Minuten, Anteil der Modulnote: 100%
Voraussetzungen für die Vergabe von Kreditpunkten: 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
Verwendung des Moduls (in anderen Studiengängen): Studiengang Maschinenbau (Bachelor), Studiengang SEPM (Bachelor)
Stellenwert der Note für die Endnote: 6 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Detlef Gerhard

Fertigungsautomatisierung					
Modul-Nr: 136460	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Aufbauend auf den im Bachelor-Studiengang vermittelten Kenntnissen überAutomatisierungstechnik lernen die Studierenden vertiefte ingenieurwissenschaftlicheGrundlagen im Bereich der automatisierten Fertigungsverfahren mit bahngesteuertenWerkzeugen kennen. Ein Schwerpunkt wird dabei auf die NC- und Robotersteuerungenund deren Programmierung gelegt. Ein zweiter Schwerpunkt befasst sich mit vernetztenFertigungssystemen.</p> <ul style="list-style-type: none"> • Die Studierenden können die erlernten Kenntnisse auf andere maschinenbaulicheProblemmstellungen übertragen und somit die Automatisierungspotentiale innovativerFertigungsverfahren beurteilen. • Sie können die Technologie moderner NC- Steuerungen aufgabenspezifisch anwenden und Trends der Steuerungstechnik erkennen. • Weiterhin haben die Studierenden vertiefte, interdisziplinäre Methodenkompetenzerworben und können diese situativ anpassen. • Die Studierenden werden in die Lage versetzt, die Problematik derKoordinatentransformation bei Industrierobotern darzustellen und numerische Lösungswege anzuwenden. • Sie können die Einflussgrößen auf die Fertigungsgenauigkeit erkennen und die verschiedenen Arten der Genauigkeit unterscheiden. • Darüber hinaus werden Kompetenzen zu den Anwendungspotentialen der Feldbus-und Internettechnologie als Bestandteil moderner Fertigungssysteme vermittelt <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten <p>Fachspezifische Kompetenzen: Technologien:</p>					
Inhalt:					

KAPITEL 7. INGENIEURINFORMATIK

In der Veranstaltung werden zunächst unterschiedliche Fertigungsverfahren vorgestellt, um deren spezifischen Anforderungen an die Automatisierung herauszuarbeiten. Im Fokus stehen hierbei innovative Fertigungsverfahren wie das Rapid-Prototyping, die Hochgeschwindigkeitszerspanung, die inkrementelle Umformung oder die Laserbearbeitung. Im Abschnitt NC-Steuerungen werden die Datenaufbereitung, die Bahnsteuerungsfunktionen mit Geschwindigkeitsführung, Interpolation und Koordinatentransformation sowie die Lageregelung behandelt. Es werden Entwicklungspotentiale in Richtung offene NC-Steuerungen und STEP-NC aufgezeigt. In Abschnitt Robotersteuerungen werden insbesondere die spezifischen Probleme und Lösungen der Transformation vom Effektor- zum Basiskoordinatensystem vorgestellt. Die für Werkzeugmaschinen und Roboter wichtigen Wegmesssysteme werden in ihrer Funktionsweise erläutert. Ein weiterer Abschnitt behandelt das Thema Genauigkeit und stellt die für NC-Werkzeugmaschinen und Roboter zu berücksichtigenden Normen vor. Die wichtigen Feldbusse PROFIBUS und INTERBUS sowie die Sensor-/Aktorbusse CAN und SERCOS werden in Aufbau und Kommunikationsstruktur eingehend vermittelt und die Potentiale der Intertechnik in Steuerungsanwendungen behandelt. Im Abschnitt sicherheitsgerichtete Steuerungen werden die relevanten Konzepte für SPS- Sicherheitssteuerungen und sichere Feld- und Sensor- Aktorbusse dargestellt.

Lehrformen:

Vorlesung (+Übung)

Prüfungsformen:

Klausur, 120 Minuten

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr.-Ing. Bernd Kuhlenkötter

Menschenzentrierte Robotik					
Modul-Nr: 135040	Credits: 6 CP	Workload: 180 h	Semester: 5. Sem.	Turnus: jedes WiSe	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung mit Projektarbeit			Kontaktzeit: 2 SWS	Selbststudium: 150 h	Gruppen- größe: ca. 35
Teilnahmevoraussetzungen: Für den Kurs sollten die Studierenden Teamfähigkeit mitbringen und Interesse an interdisziplinären Themen haben, die über den ingenieurwissenschaftlichen Bereich hinausgehen (wie z.B. die psychologische Implikationen der Robotik).					
Lernziele (learning outcomes): Zielsetzung:					
<ul style="list-style-type: none"> • Alle Studierenden haben Grundkenntnisse über die Entwicklungen, Anwendungsbereiche und die aktuellen Trends im Bereich der Mobilen und Servicerobotik. • Die Studierenden der Ingenieurwissenschaften verstehen die Funktionsweise und den Aufbau von Robotersystemen und sind in der Lage diese zu programmieren • Sie haben Grundkenntnisse über Forschungsmethoden der Mensch-Roboter-Interaktion, Mensch-Roboter-Kollaboration und sind in der Lage Gestaltungsempfehlungen auf Basis empirischer Befunde abzuleiten. • Alle Studierenden sind in der Lage die multimediale Landschaft zur Kommunikation zwischen Roboter und Mensch differenziert zu betrachten. • Sie haben die Fähigkeit kleinere Projekte selbständig innerhalb einer Gruppe zu planen und durchzuführen. 					
Kompetenzen:					
<ul style="list-style-type: none"> • Die Studierenden verfügen über fachübergreifende Methodenkompetenz. • Sie erlernen die Arbeit in interdisziplinären Teams. • Sie können Erkenntnisse/Fertigkeiten auf konkrete maschinenbauliche/ingenieurwissenschaftliche Problemstellungen übertragen. • Sie lernen Grundlagen anderer Disziplinen im interdisziplinären Austausch kennen. • Sie verfügen über experimentelles Fachvokabular, kennen empirische Forschungsmethoden und die Grundlagen der Sozialpsychologie. • Sie können disziplinübergreifende Inhalte aus nicht technischen Bereichen reflektieren und verantwortungsbewusst neue Ansätze in den Projektarbeiten entwickeln. 					

- Sie können durch die Projektarbeiten effektiv und effizient in Teams kommunizieren, diskutieren und ihre Arbeiten im Anschluss präsentieren

Inhalt:

Die Studierenden werden in fachübergreifenden Gruppen an konkreten Problemstellungen im Bereich der menschenzentrierten Robotik arbeiten. Zur Gestaltung des sozio-technischen Systems aus Mensch(en) und Roboter(n), werden sowohl ingenieurwissenschaftliche als auch psychologische Fragen berücksichtigt.

Vorbereitend hierfür wird aus technischer Sicht eine thematische Einführung in die Historie, Anwendungsfeldern und Funktionsweisen von Robotersystemen gegeben. Dabei wird vor allem auf die mobile Servicerobotik und die Mensch-Roboter-Kollaboration eingegangen. Zur menschengerechten Gestaltung der Interaktion mit der Roboterplattform, wird eine Einführung in psychologische Effekte der Mensch-Technik-Interaktion gegeben, sowie die soziale Robotik und ihre Anwendungsfelder vorgestellt.

Auf dieser Basis bearbeiten die Studierenden dann in interdisziplinären Gruppen individuelle Problemstellungen unter wissenschaftlichen Gesichtspunkten zur Weiterentwicklung einer mobilen Roboterplattform. Hierbei steht thematisch die erfolgreiche Kommunikation und Interaktion zwischen Mensch und Roboter (und der Eindruck des Roboters auf dem Menschen) im Vordergrund. Es besteht die Möglichkeit die ausgearbeiteten Lösungsansätze zu implementieren und somit eine Validierung des Konzeptes durchzuführen.

Die Studierenden werden bei der Projektplanung und dem Projektmanagement unterstützt, indem ihnen die Grundlagen des Projektablaufs für die jeweiligen Projektphasen vermittelt werden. Während der Projektlaufzeit wird durch Zwischengespräche die Projektentwicklung überprüft und reguliert. So werden die in der Lehrveranstaltung vorgestellten Methoden und das erlernte Wissen praktisch angewendet und das Arbeiten in interdisziplinären Projektgruppen eingeübt.

Lehrformen:

Vorlesung mit Projektarbeit

Prüfungsformen:

Prüfung: Projektarbeit - Eigenständige Projektarbeit mit Zwischengesprächen zum Zielabgleich - Anteil der Modulnote: 20 %

Prüfung: Dokumentation und Präsentation - Anfertigung einer kurzen Dokumentation und Präsentation der Ergebnisse. - Anteil der Modulnote: 80 %

Voraussetzungen für die Vergabe von Kreditpunkten:

- Bestandene Modulabschlussprüfung: Dokumentation und Präsentation
- Teilnahme an allen Zwischengesprächen
- Bestandene Modulabschlussprüfung: Dokumentation und Präsentation

Verwendung des Moduls (in anderen Studiengängen): Das Modul ist als interdisziplinäres Wahlmodul sowohl in den Ingenieurwissenschaften als auch in der Psychologie wählbar.

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr.-Ing. Bernd Kuhlenkötter

Prof. Dr.-Ing. Bernd Kuhlenkötter, Dr. Laura Hoffmann

Bioinformatik

Grundlagen der Bioinformatik					
Modul-Nr: 190533 (Übung: 190543)	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine					
Empfohlene Vorkenntnisse: Grundkenntnisse in Biologie					
Lernziele (learning outcomes): Die Teilnehmer/innen erwerben Grundkenntnisse und erhalten einen Einblick in die aktuellen Werkzeuge und zugrunde liegenden Methoden der Bioinformatik. Erworbene Kompetenzen liegen vor allem im Bereich des Erlernens bioinformatischer Werkzeuge, des Identifizierens angemessener Bioinformatik Methoden für biologische Fragestellungen sowie das Erlernen von formalem mathematisch-informatischen Denkens. Hierbei spielt das Erlernen interdisziplinären Denkens und das Anwenden von Fähigkeiten und Wissen über Fächergrenzen hinweg eine besondere Rolle.					
Vermittelte Kompetenzen: Kernkompetenzen:					
<ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten 					
Inhalt: Bioinformatische Werkzeuge und Methoden sind zu einem festen Bestandteil der biologischen Forschung geworden. Die Vorlesung gibt einen Überblick über die aktuellen Werkzeuge und die zugrunde liegenden Methoden, mit einem besonderen Schwerpunkt der Analyse von Sequenz und Struktur von Proteinen. In den Übungen wird einerseits die Anwendung dieser Werkzeuge in der Praxis vermittelt, andererseits die theoretischen Grundlagen anhand von Übungsaufgaben vertieft.					

KAPITEL 8. BIOINFORMATIK

Lehrformen: Vorlesung (+Übung)
Prüfungsformen: Klausur
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: PD Dr. Mathias Lübben, Prof. Dr. Axel Mosig PD Dr. Mathias Lübben, Prof. Dr. Axel Mosig, Prof. Dr. Raphael Stoll
Sonstige Informationen: Literatur: A. Lesk, Introduction to Bioinformatics, Oxford University Press, 2002. Biologische Grundkenntnisse können ggf. durch entsprechende Online-Kurse erworben werden.

Methoden der Bioinformatik					
Modul-Nr: 190502 (Übung: 190522)	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine					
Empfohlene Vorkenntnisse: Grundkenntnisse in Biologie					
<p>Lernziele (learning outcomes): Die Studierenden erlernen interdisziplinäre Denkweisen und notwendige Grundkenntnisse, um aktuelle Forschungsthemen der Bioinformatik verfolgen zu können (Vorlesung). Anhand von Fragestellungen der Biologie werden Fähigkeiten des algorithmischen und statistischen Modellierens und Problemlösens erworben (Übungsaufgaben).</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten • Selbständiges Lernen und Arbeiten <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Algorithmik • Graphentheorie • Mathematik • Optimierungsmethoden 					
<p>Inhalt:</p> <ul style="list-style-type: none"> • Algorithmische und statistische Grundlagen der Bioinformatik (Reguläre Ausdrücke, Endliche Automaten, Turing Maschinen, Komplexität, Dynamische Programmierung, Maximum Likelihood, Hidden Markov Modelle, Poisson Prozesse) • Algorithmen zur Analyse von Sequenz und Struktur von Bio-Molekülen; Rekonstruktion evolutionärer Beziehungen zwischen Sequenzen und Strukturen 					
<p>Lehrformen: Vorlesung (+Übung)</p>					

KAPITEL 8. BIOINFORMATIK

Prüfungsformen: Klausur
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Axel Mosig
Sonstige Informationen: Literatur: <ol style="list-style-type: none">1. R. Durbin, S. Eddy, A. Krogh, G. Mitchinson, Biological Sequence Alignments, Cambridge University Press, 2004.2. N. Jones, P. Pevzner, An Introduction to Bioinformatics Algorithms, MIT Press, 2004.3. P. Pevzner, R. Shamir, Computing Has Changed Biology—Biology Education Must Catch Up, Science 325(5940):541-542, 2009.4. T.W. Tan, S.J. Lim, A.M. Khan, S. Ranganathan, A proposed minimum skill set for university graduates to meet the informatics needs and challenges of the "-omics" era, BMC Genomics 10(Suppl 3):S36, 2009. <p>Biologische Grundkenntnisse können ggf. durch entsprechende Online-Kurse erworben werden.</p>

Computerlinguistik

Einführung in die Linguistik					
Modul-Nr: MISSING	Credits: 6 CP	Workload: 180 h	Semester: 3./5. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Grundkurs Einführung in die Linguistik			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppen- größe: 90
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Englischkenntnisse B2, Mathematikkennntnisse, Informatikkennntnisse					
<p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über grundlegendes Wissen und elementare analytische/methodische Fertigkeiten in den zentralen Teildisziplinen der Linguistik, die Sie kennenlernen – Phonetik/Phonologie, Morphologie, Syntax, Semantik, Pragmatik. Sie können</p> <ul style="list-style-type: none"> • die grundlegenden Begriffe und Konzepte der Phonetik/Phonologie, Morphologie, Syntax, Semantik und Pragmatik auflisten, dabei terminologisch korrekt benennen und Sie können die genannten Konzepte auch darstellen. • Die grundlegende Terminologie der im Modul behandelten linguistischen Teildisziplinen sowie verschiedener Schulen und zentraler Forschungsgebiete der Linguistik sind Ihnen sowohl auf Deutsch als auch auf Englisch bekannt und Sie können sprachwissenschaftliche Sachverhalte und Prozesse zutreffend bezeichnen, aufzählen und definieren. • Sie verstehen somit die linguistischen Fachbegriffe der o. a. Teildisziplinen (insbesondere auch in englischsprachiger Literatur, die für uns der Normalfall ist) und können sie auch richtig anwenden. • Sie können die elementaren analytischen Methoden der strukturellen Linguistik in Phonetik/Phonologie, Morphologie, Syntax, Semantik und Pragmatik identifizieren und die entsprechenden Prozeduren skizzieren. 					

- Folglich können Sie auch elementare Darstellungen und Analysen klassifizieren und den verschiedenen linguistischen Teildisziplinen und Forschungsgebieten zuordnen.
- Sie können erste, einfache Analysen sprachlicher Daten auf den Beschreibungsebenen der Phonetik/Phonologie, Morphologie, Syntax, Semantik und Pragmatik durchführen und können dafür Methoden je nach analytischer Aufgabenstellung richtig auswählen, zuordnen und natürlich anwenden.

Inhalt:

Dieser Kurs vermittelt die grundlegenden Konzepte und Strukturmerkmale der zentralen linguistischen Teildisziplinen bzw. Beschreibungsebenen (Phonetik/Phonologie, Morphologie, Syntax, Semantik, Pragmatik):

Phonetik/Phonologie:

Die Phonologie untersucht lautliche Struktur/en von Sprachen. Dabei befasst sie sich u. a. mit den für ihre Beschreibung und Analyse relevanten wichtigsten physiologischen und physikalischen Gegebenheiten. In diesem Grundkurs lernen Sie zunächst phonologische Methoden und Analysen sowie grundlegendes Begriffsinventar und wesentliche Analysemethoden u. a. in den folgenden Bereichen kennen:

- Sprachlaute vs. Schrift (Transkription, IPA);
- artikulatorische Parameter, Lautklassifikation und segmentale Komposition;
- phonemische Analyse, phonologische Regeln;
- Silben und Silbifizierung.

Morphologie:

Die Morphologie untersucht primär die interne Struktur von Wörtern. Sie lernen morphologische Methoden und Analysen kennen sowie grundlegendes Begriffsinventar und wesentliche Analysemethoden. Es geht dabei u. a. um Betrachtung, Diskussion und Anwendung unterschiedlicher Aspekte und Methoden morphologischer Beschreibung und Analyse. Behandelt werden dabei u. a.

- Morphembegriff, Affigierung, Allomorphie;
- Eigenschaften von Derivation und Flexion in den Sprachen der Welt;
- formale Typen morphologischer Operationen.

Syntax:

Syntax beschäftigt sich mit der Kombination von Wörtern zu komplexeren Einheiten bis hin zu Sätzen. Sie untersucht dabei, welche Wörter mit welchen anderen kombiniert werden können, welche Stellungsregularitäten dabei auftreten und wie die resultierende Struktur mit der Bedeutung solcher Einheiten zusammenhängt. In diesem thematischen Teilbereich des Grundkurses lernen Sie grundlegende Konzepte der Syntax kennen, darunter u. a.

- Wortarten, syntaktische Kategorien und Funktionen;
- semantische Rollen, Valenz, Konstituenz, Dependenz.

KAPITEL 9. COMPUTERLINGUISTIK

Semantik:

Semantik ist die teildisziplin der Linguistik, das sich mit den Bedeutungen sprachlicher Ausdrücke beschäftigt. Ein zentrales Thema ist folglich die Beantwortung der Frage, was man unter Bedeutung eines sprachlichen Ausdrucks versteht. Dabei lernen Sie grundlegende Bedeutungsarten kennen und beschäftigen sich zunächst mit den Bedeutungen von Wörtern, sprich der lexikalischen Semantik. Ferner bekommen Sie einen Überblick über Bedeutungsphänomene auf Satzebene. Die Themen dieses Grundkurses sind u. a.:

- Bedeutungsarten: deskriptive, soziale, expressive Bedeutung;
- Ausdrucks-, Äußerungsbedeutung und kommunikativer Sinn;
- Referenz und Proposition;
- semantische Relationen, Polysemie, Metonymie, Metapher.

Pragmatik:

Pragmatik befasst sich schwerpunktmäßig mit der Bedeutung sprachlicher Ausdrücke im (situativen) Kontext. Die Themen für diesen Teilbereich des Grundkurses sind Betrachtung, Diskussion und Anwendung von Konzepten und Methoden der Pragmatik, Sprachgebrauch und Bedeutungsaspekte der Sprachverwendung, darunter u. a.

- Interaktion zwischen Pragmatik und Semantik;
- Konversationsmaximen und Implikaturen;
- Präsuppositionen, Deixis, Sprechakte.

Lehrformen:

Die Lehrform ist aufgrund der Gruppengröße vermittlungsorientiert, d. h.

- Präsentation durch Kursleiter/in mit Fragen an die Teilnehmer/innen und Diskussion einzelner Aspekte, sowie
- kleinere Übungen bzw. Analysen während der Sitzungen, die Sie z. B. mit Ihren Sitznachbarn zusammen durchführen.

Tutorien: Besonders wichtig ist hier für Sie, dass der Grundkurs von einem Tutorium begleitet wird. Somit haben Sie zusätzlich zum Unterricht durch die Lehrenden im Hörsaal noch die Möglichkeit, sich den Stoff angeleitet von Tutor/inn/en (das sind erfahrene und fachlich versierte Kommiliton/inn/en) in sehr kleinen Gruppen weiter zu erarbeiten, nachzuarbeiten und durch weitere Übungen zu festigen.

Prüfungsformen:

Die Modulprüfung wird durch eine Modulabschlussklausur abgelegt, die in der Regel gegen Vorlesungsende stattfindet. Sie erhalten zu Beginn der Vorlesungszeit einen Überblick über die verschiedenen Leistungen, die Sie für die Kreditierung des Moduls erbringen müssen (z. B. zu Art und Anzahl der Hausaufgaben (vgl. unten), Form, Dauer und Terminierung der Klausur).

Voraussetzungen für die Vergabe von Kreditpunkten:

Das Modul wird Ihnen mit insgesamt 5,0 CP kreditiert, die sich aus den CP für den Grundkurs (4 CP) sowie den Workload für die Modulprüfung (1 CP) zusammensetzen. In diesem Modul müssen Sie (a) schriftliche Hausaufgaben in dem Grundkurse' im Verlauf der Vorlesungszeit bearbeiten, die als Studienleistungen kreditiert werden, und (b) eine Modulabschlussklausur als Modulprüfung am Ende der Vorlesungszeit bestehen. Das Modul wird Ihnen erst dann als bestanden bewertet und entsprechend kreditiert, wenn Sie (a) die Studienleistungen des Grundkurses erbracht haben sowie (b) auch die Modulprüfung (sprich die Modulabschlussklausur) bestanden haben.

Verwendung des Moduls (in anderen Studiengängen): Ist Teil des Moduls „Grundlagen der Linguistik“ im Studiengang Computer-, Psycho- und Theoretische Linguistik

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Dr. Martin Hoelter

Prof. Dr. Eva Belke, Prof. Dr. Stefanie Dipper, Dr. Martin Hoelter, Prof. Dr. Ralf Klabunde, Prof. Dr. Agata Renans.

Sonstige Informationen:

Eine Anmeldung zur Veranstaltung ist über eCampus jeweils bis zum 01.10. vorzunehmen. Die Veranstaltung kann parallel zu „Methoden der Computerlinguistik“ belegt werden.

Methoden der Computerlinguistik					
Modul-Nr: MISSING	Credits: 5 CP	Workload: 150 h	Semester: 3./5.. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Semiar b) praktische Übungen			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: 35
<p>Teilnahmevoraussetzungen: Teilnahme am Modul Computerlinguistik 1</p> <p>Empfohlene Vorkenntnisse: Grundlegendes Wissen über die wichtigsten basalen Konzepte und Methoden aus der Theoretischen Linguistik und der Programmierung.</p>					
<p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls verfügen Sie über grundlegendes Wissen über Computerlinguistik:</p> <ul style="list-style-type: none"> • Sie kennen die wichtigsten, grundlegenden Analyseebenen in der Computerlinguistik und ihre spezifischen Eigenschaften. • Sie kennen verschiedene, insbesondere statistische Verfahren zur automatischen Analyse von Sprachdaten sowie die dazugehörigen effizienten Algorithmen, können diese implementieren und auf eine eigene Fragestellung anwenden. • Sie kennen die spezifischen Probleme bei der automatischen Verarbeitung sprachlicher Daten (wie z. B. sparse data, Ambiguitäten) und können diese anhand konkreter Beispiele erklären. • Sie kennen eine Reihe von Evaluationsverfahren sowie ihre Vor- und Nachteile und können diese anwenden. • Sie kennen relevante Ressourcen und wissen, in welcher Form sie eingesetzt werden. 					
<p>Inhalt: Dieses Modul führt in die Kernmethoden und -modelle der computerlinguistischen Forschung ein. Neben theoretischen Grundlagen der Computerlinguistik (Chomsky-Hierarchie) werden entsprechende Modellierungen der verschiedenen Komplexitätsebenen eingeführt. Dabei werden vorrangig statistische Verfahren behandelt, bei denen das System Informationen aus Daten lernt. Neben klassischen probabilistischen Verfahren werden auch Methoden des Deep Learning behandelt, das in den letzten Jahren vermehrt Anwendung in der maschinellen Sprachverarbeitung findet. Darüber hinaus soll das Modul die Teilnehmer/innen befähigen, sich Fachpublikationen zu computerlinguistischen Arbeiten von der Fragestellung über die Umsetzung bis hin zu den Ergebnissen zu erschließen und für Präsentationen und schriftliche Arbeiten angemessen aufzubereiten.</p>					

KAPITEL 9. COMPUTERLINGUISTIK

<p>Lehrformen: Seminar mit Anteilen von Inverted Classroom Settings mit digital verfügbaren, primär videobasierten Lehrmaterialien zum Selbststudium. Praktische Programmierübungen.</p>
<p>Prüfungsformen: Implementation mit Dokumentation als benotete Modulprüfung.</p>
<p>Voraussetzungen für die Vergabe von Kreditpunkten: Erfolgreicher Abschluss der Modulprüfung sowie bestandene Studienleistungen in (a) und (b): Portfolios aus schriftlichen Hausaufgaben/Übungen, Präsentationen, Diskussionsbeiträgen. Gesamtnote: Die Gesamtnote des Moduls ergibt sich im Regelfall nur aus der Note der Modulprüfung. Das Modul wird aber erst dann als bestanden bewertet und entsprechend kreditiert, wenn die Modulprüfung sowie die obligatorischen Studienleistungen insgesamt bestanden sind.</p>
<p>Verwendung des Moduls (in anderen Studiengängen): Ist als Modul „Aufbaumodul Computerlinguistik“ Betsandteil im Studiengang Computer-, Psycho- und Theoretische Linguistik</p>
<p>Stellenwert der Note für die Endnote: 5 / 175</p>
<p>Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Ralf Klabunde Prof. Dr. Stefanie Dipper, Prof. Dr. Ralf Klabunde</p>
<p>Sonstige Informationen: Eine Anmeldung zur Veranstaltung ist über eCampus jeweils bis zum 01.10. vorzunehmen. Die Veranstaltung kann parallel zu Einführung in die Linguistik belegt werden.</p>

Kryptologie

Diskrete Mathematik II					
Modul-Nr: 150310 (Übung: 150311)	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 3 SWS	Selbststudium: 135 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Keine					
<p>Lernziele (learning outcomes): Die Vorlesung gibt eine Einführung in die Codierungstheorie und in die Theorie der Berechenbarkeit.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Algorithmik • Graphentheorie • Mathematik • Optimierungsmethoden 					
<p>Inhalt:</p> <ul style="list-style-type: none"> • Eindeutig entschlüsselbare Codes • Kompakte und optimale Codes • Lineare und duale Codes 					

KAPITEL 10. KRYPTOLOGIE

- Turingmaschine
- Komplexitätsklassen P und NP
- Polynomielle Reduktion
- Quadratische Reste

Lehrformen:

Vorlesung (+Übung)

Prüfungsformen:

Klausur

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Alexander May

Sonstige Informationen:

Im Studiengang ITS läuft die Vorlesung unter dem Titel "Einführung in die theoretische Informatik".

Zahlentheorie					
Modul-Nr: 150232 (Übung: 150233)	Credits: 9 CP	Workload: 270 h	Semester: 3-5. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: 6 SWS	Selbststudium: 180 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Lineare Algebra					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • kennen Studierende typische Fragestellungen der elementaren Zahlentheorie, • haben Studierende einen Einblick in die Ästhetik der ganzen und rationalen Zahlen, • verstehen Studierende die historische Entwicklung der Zahlentheorie und haben somit einen Einblick in die Geschichte der Mathematik im Ganzen, • besitzen Studierende Grundkenntnisse für die modernen Anwendungen der Zahlentheorie, insbesondere im Bereich der Kryptographie • haben Studierende ihr Abstraktionsvermögen geschärft, • haben Studierende ihre Fähigkeit verbessert, komplexe Probleme in Teilprobleme zu zerlegen und diese zu lösen 					
Inhalt: Division mit Rest, Zahlendarstellung ganzer Zahlen, Euklidischer Algorithmus, eindeutige Primfaktorzerlegung, Modulare Arithmetik, RSA-Verschlüsselung, chinesischer Restsatz, Struktur der primen Restklassengruppe, pythagoräische Zahlentripel, 2-Quadrate-Satz von Fermat, Gaußsches Reziprozitätsgesetz, Pellische Gleichung, Kettenbrüche, euklidische Ringe, endliche Körper, Primzahltests					
Lehrformen: Vorlesung mit begleitenden Übungsgruppen					
Prüfungsformen: schriftliche Prüfung, 120-180min					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					
Stellenwert der Note für die Endnote: 9 / 175					

Modulbeauftragte/r und hauptamtlich Lehrende:

Dr. Viktoriya Ozornova

Sonstige Informationen:

Literatur: Müller-Stach, Piontkowski, "Elementare und algebraische Zahlentheorie", Vieweg+Teubner, 2011

Einführung in die Kryptographie 1					
Modul-Nr: 141022	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Capacity for abstract and logical thought.					
<p>Lernziele (learning outcomes): The students understand essential symmetric encryption systems in practice and the basics of asymmetric cryptography. Furthermore, the students familiarize themselves with the ways of thinking in modern cryptography.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten • Selbständiges Lernen und Arbeiten <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Algorithmik • Mathematik • IT-Sicherheit 					
<p>Inhalt:</p> <p>In the beginning, the fundamental concepts of cryptography and information security are introduced, as well as some historical ciphers. Next, stream ciphers are introduced. The main part of the lecture concerns block ciphers and their applications. As important examples of symmetric cryptographic systems, the Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are introduced. As the last part of the lecture, the principles of asymmetric cryptography are introduced, in particular the widely-used RSA cryptosystem.</p> <p>In addition to cryptographic algorithms, the mathematical concepts that are needed for the algorithms are introduced, e.g., modular arithmetic, the Euclidian algorithm and finite fields.</p>					
Lehrformen: Vorlesung (+Übung)					
Prüfungsformen: Klausur, 120 Minuten					

KAPITEL 10. KRYPTOLOGIE

Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Christof Paar M. Sc. Maik Ender, B. Sc. Julian Speith
Sonstige Informationen: Literatur: C. Paar, J. Pelzl: Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2009

Einführung in die Kryptographie 2					
Modul-Nr: 141023	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Inhalte der Vorlesung "Einführung in die Kryptographie 1"</p>					
<p>Lernziele (learning outcomes): The students understand the asymmetric encryption schemes that are most relevant in practice, and how to realize security services with cryptographic primitives.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Analytisches und logisches Denken • Fächerübergreifendes Denken und Arbeiten • Selbständiges Lernen und Arbeiten <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • Algorithmik • Mathematik • IT-Sicherheit 					
<p>Inhalt: A major part of the lecture is concerned with asymmetric cryptographic algorithms. Crypto schemes based on the discrete logarithm problem are treated in detail. In particular, the Diffie-Hellman key exchange, ElGamal encryption and Elliptic Curve Cryptography (ECC) are introduced. Subsequently, schemes and protocols based on both symmetric and asymmetric primitives are developed. Specifically, digital signatures, message authentication codes (MAC), hash functions, certificates and lastly, protocols for key exchange as well as security services.</p>					
<p>Lehrformen: Vorlesung (+Übung)</p>					
<p>Prüfungsformen: Klausur, 120 Minuten</p>					
<p>Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur</p>					

Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Christof Paar M. Sc. Maik Ender, B. Sc. Julian Speith
Sonstige Informationen: Videomitschnitte unter www.crypto-textbook.com Literatur: C. Paar, J. Pelzl: Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2009

Kryptographie auf hardwarebasierten Plattformen					
Modul-Nr: 141031	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung (+Übung)			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
<p>Teilnahmevoraussetzungen: Keine</p> <p>Empfohlene Vorkenntnisse: Empfohlen: Grundlagen der Kryptographie und Datensicherheit, Basiswissen Digitaltechnik</p>					
<p>Lernziele (learning outcomes): Die Studierenden kennen die Konzepte der praxisnahen Hardware- entwicklung mit abstrakten Hardwarebeschreibungssprachen (VHDL) und die Simulation von Hardwareschaltungen auf FPGAs. Sie beherrschen Stan- dardtechniken der hardwarenahen Prozessorentwicklung und sind zur Im- plementierung von symmetrischen und asymmetrischen Kryptosystemen auf modernen FPGA-Systemen in der Lage.</p> <p>Vermittelte Kompetenzen:</p> <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Fächerübergreifendes Denken und Arbeiten <p>Fachspezifische Kompetenzen:</p> <ul style="list-style-type: none"> • IT-Sicherheit • Algorithmik • Mathematik 					
<p>Inhalt:</p> <p>Kryptographische Systeme stellen aufgrund ihrer Komplexität ins- besondere an kleine Prozessoren und eingebettete Systeme hohe Anforde- rungen. In Kombination mit dem An- spruch von hohem Datendurchsatz bei geringsten Hardwarekosten ergeben sich hier für den Entwickler grundlegende Probleme, die in dieser Vorlesung beleuchtet werden sollen. Die Vor- lesung behandelt die interessantesten Aspekte, wie man aktuelle kryptographische Verfahren auf praxisnahen Hardwaresystemen implemen- tiert. Dabei werden Kryptosysteme wie die Blockchiffre AES, die Hashfunk- tionen SHA-1 sowie asymmetrische Systeme RSA und ECC behandelt. Wei- terhin werden auch spezielle Hardwareanforderungen wie beispielsweise der Erzeugung echten Zufalls (TRNG) sowie der Einsatz von Physically Unclo- nable Functions (PUF) besprochen. Die effiziente Implementierung dieser Kryptosysteme, insbesondere in Be- zug auf die Optimierung für Hochgeschwindigkeit, wird auf modernen FPGAs besprochen und in praktischen Übungen mit Hilfe der Hardwarebeschrei- bungssprache VHDL umge- setzt. Vorlesungsbegleitend wird ein Moodle-Kurs angeboten, der zusätzliche Inhalte sowie die praktischen Übungen bereithält.</p>					

KAPITEL 10. KRYPTOLOGIE

Lehrformen: Vorlesung (+Übung)
Prüfungsformen: Klausur, 120 Minuten
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr.-Ing. Tim Güneysu

Logik in der Informatik					
Modul-Nr: 150345	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: jedes WiSe	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung mit begleitenden Übungen			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Vorlesung über Theoretische Informatik					
Lernziele (learning outcomes): Die Studierenden lernen wie sich Problemstellungen durch geeignete logische Systeme modellieren lassen. Sie sollen Syntax und Semantik verschiedener logischer Systeme beherrschen und nutzen können. Sie sollen einige klassische logische Kalküle und Algorithmen kennen sowie diese durchführen können. Sie sollen weiterhin ein grundlegendes Verständnis für die Logik-Programmierung entwickeln und insbesondere einfache Sachverhalte durch Prolog-Programme auszudrücken können.					
Inhalt: Logische Methoden spielen in vielen modernen Anwendungen der Informatik eine wichtige Rolle. Aus Datenbanken werden relevante Informationen mit Hilfe auf Logik basierender Anfragesprachen extrahiert; die formale Verifikation von Software und Hardware basiert auf logischen Spezifikationsprachen und Algorithmen für diese; und Methoden für das automatisierte Schlussfolgern in der künstlichen Intelligenz haben ihre Grundlage in der formalen Logik. In dieser Veranstaltung werden die formalen Grundlagen von modernen Logiken behandelt, mit einem Fokus auf ihrer Anwendung in der Informatik. Neben der klassischen Aussagenlogik und Prädikatenlogik betrachten wir auch Modallogik. Für jede dieser Logiken formalisieren wir Syntax und Semantik, lernen wie sich informatische Szenarien in ihnen modellieren lassen, und betrachten Algorithmen und Kalküle für Unerfüllbarkeit und Folgerungsbeziehung.					
Lehrformen: Vorlesung mit begleitenden Übungen					
Prüfungsformen: Abschlussprüfung; mündliche Prüfung (20-30min) oder schriftliche Klausur (120min) in Abhängigkeit der Teilnehmerzahl					
Voraussetzungen für die Vergabe von Kreditpunkten: bestandene mündliche oder schriftliche Prüfung					
Verwendung des Moduls (in anderen Studiengängen): Studiengang Mathematik					
Stellenwert der Note für die Endnote: 5 / 175					
Modulbeauftragte/r und hauptamtlich Lehrende: Jun.-Prof. Thomas Zeume					

Sonstige Informationen:

Erst wieder im WS 22/23

Literatur: Einstiegsliteratur für diese Veranstaltung sind die Bücher:

- M. Kreuzer and S. Kühling. Logik für Informatiker. Pearson, 2006
- Uwe Schöning. Logik für Informatiker. Spektrum Akademischer Verlag, 2000

Automata theory					
Modul-Nr: 150334	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Introductory course on theoretical computer science					
Lernziele (learning outcomes): Students learn about different automata models that are used in computer science. They learn how models can be analysed with respect to closure properties and algorithmic properties. They shall develop an understanding of the power of distinct automata models, and be enabled to develop and analyse new automata models.					
Inhalt: Automata play an important role in computer science and its applications. As an example, finite state automata as introduced in introductory courses on theoretical computer science, are used in compiler construction and in pattern matching for strings. In this course we systematically study the theoretical foundations of diverse automata models and establish connections of automata theory to other areas such as logic and algebra. Automata models have been developed for a plethora of applications, among other we will study <ul style="list-style-type: none"> • w-Automata: Very similar to finite state automata, these automata work on infinite words. They are used in formal verification of hardware and software. • Tree automata: Inputs for these automata are trees and they are used for instance in specification and querying of tree-shaped data, as for instance XML or JSON. • Probabilistic automata: These automata accept their inputs with certain probabilities and can be used in pattern recognition and formal verification. <p>The focus of this course is on theoretical properties of automata, but we will also consider some applications.</p>					
Lehrformen: Vorlesung					
Prüfungsformen: oral exam					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					

Verwendung des Moduls (in anderen Studiengängen): Diese Veranstaltung richtet sich an Studierende der Mathematik, der Informatik, der Angewandten Informatik und der ITS.
Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Thomas Zeume
Sonstige Informationen: Nicht im SS 22! Im SoSe 2020 zunächst 2+2, in Zukunft ggf. 4+2 Literatur: <ul style="list-style-type: none">• Bakhadyr Khoussainov and Anil Nerode. Automata theory and its applications, volume 21. Springer Science and Business Media, 2012• Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Christof Löding, Denis Lugiez, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications, 1997• Wolfgang Thomas, Thomas Wilke, et al. Automata, logics, and infinite games: a guide to current research, volume 2500. Springer Science and Business Media, 2002

Neuroinformatik

Introduction to Artificial Intelligence					
Modul-Nr: 310502 (Übung: 310512)	Credits: 5 CP	Workload: 150 h	Semester: 2. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung Introduction to Artificial Intelligence			Kontaktzeit: 4 SWS	Selbststudium: 90 h	Gruppen- größe: ca. 300
Teilnahmevoraussetzungen: Basic knowledge of calculus and linear algebra.					
Lernziele (learning outcomes): After successful completion of this course, students will be able to <ul style="list-style-type: none"> • summarize a number of fundamental methods in artificial intelligence, • explain their mathematical basis and algorithmic nature, • apply them to simple problems, • decide which methods are suitable for which problems, and • communicate about the above aspects in English. 					
Inhalt: This course gives an overview over representative methods in artificial intelligence: formal logic and reasoning, classical AI methods, cognitive modeling, computational neuroscience, deep neural network, machine learning, natural language processing, probabilistic reasoning, robotics.					
Lehrformen: The course consists of a lecture, which provides the content, and tutorial, which deepens the understanding of the content. The course is taught by a team of lecturers from different subfields of artificial intelligence.					
Prüfungsformen: Abschlussprüfung; The course is concluded with a written exam (2h)					

Voraussetzungen für die Vergabe von Kreditpunkten:

Passing grade on final written exam

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Laurenz Wiskott

Laurenz Wiskott, Tobias Glasmachers, Gregor Schöner, Sen Cheng, and others

Computersehen: Einführung					
Modul-Nr: 310505 (Übung: 310515)	Credits: 5 CP	Workload: 150 h	Semester: 4 oder 6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung und Übungen Computersehen: Einführung			Kontaktzeit: 2 + 1	Selbststudium: 1 h	Gruppen- größe: 40 - 60 Studie- rende
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Ziel dieser Vorlesung ist es, die Grundlagen der digitalen Bildverarbeitung zu vermitteln. Die Studierenden sollen die Grundideen der verschiedenen Methoden verstehen und in die Lage versetzt werden, eigenständig Bildverarbeitungsalgorithmen in einer Programmiersprache ihrer Wahl umzusetzen. Vollständigkeit kann kein Ziel einer einführenden Vorlesung sein. Ein wesentliches Lernziel ist daher die Selbsterschließbarkeit moderner oder auch etablierter Bildverarbeitungsverfahren mit den Methoden, Konzepten und Herangehensweisen der Veranstaltung.</p>					
<p>Inhalt:</p> <ul style="list-style-type: none"> • Aufbau einer Kamera: Lochblendenmodell und moderne Kameras • Grundlegende Bildverarbeitungsoperationen: Punktoperatoren, Filter, Geometrische Transformationen • Punktdetektion und Punktdeskriptoren: SIFT, SURF, FAST • Bildregistrierung • Stereo-Sehen: Dynamische Programmierung und Semi-Global Matching • Einführung in das Maschinelle Lernen: Bildklassifikation • Echtzeitfähige Detektion: Viola-Jones-Detektor und Soft-Cascade 					
<p>Lehrformen: 2 stündiges Vorlesungsformat mit zusätzlich wöchentlich zu bearbeitenden praktischen Übungen in Form von mathematischen Herleitungen, Verständnisfragen und Programmimplementierungen</p>					
<p>Prüfungsformen: Klausur von 90 – 120 Minuten Länge mit mathematischen Herleitungen und Verständnisfragen</p>					
<p>Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur</p>					

KAPITEL 11. NEUROINFORMATIK

Stellenwert der Note für die Endnote: 5 / 175
Modulbeauftragte/r und hauptamtlich Lehrende: Jun.-Prof. Sebastian Houben

Artificial Neural Networks					
Modul-Nr: 310002 (Übung: 310012)	Credits: 6 CP	Workload: 180 h	Semester: 3-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Übung			Kontaktzeit: 3 SWS	Selbststudium: 120 h	Gruppen- größe: a) 150 b) 20
<p>Teilnahmevoraussetzungen: calculus, linear algebra, statistics, programming</p> <p>Empfohlene Vorkenntnisse: Grundkenntnisse in der Infinitesimalrechnung, linearen Algebra, Statistik und Informatik. Erfahrung mit einer höheren Programmiersprache.</p>					
<p>Lernziele (learning outcomes): Die mathematischen Grundlagen, Möglichkeiten und Beschränkungen überwachter Lernverfahren für Regression und Klassifikation mit künstlichen neuronalen Netzen (KNN), sowie für deren Anwendung erforderliche praktische Kenntnisse werden vermittelt. Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • verstehen Studierende die theoretisch-mathematischen Grundlagen von KNN im Kontext des überwachten Lernens. • können Studierende selbstständig zwischen verschiedenen KNN unterscheiden und in einer Anwendungssituation das geeignete Verfahren auswählen. • können Studierende grundlegende Verfahren selbstständig in einer höheren Programmiersprache implementieren, sowie ihre eigene Implementierung und Standard- Implementierung anderer auf Daten anwenden. • können Studierende Ergebnis der KNN selbstständig interpretieren, insbesondere erkennen, wann sie unrealistisch sind. <p>Kernkompetenzen:</p> <ul style="list-style-type: none"> • Selbständiges Lernen und Arbeiten • Analytisches und logisches Denken • Analyse und Interpretation von Daten • Kommunikation von quantitativen Zusammenhängen <p>Fachspezifische Kompetenzen:</p>					

KAPITEL 11. NEUROINFORMATIK

- Verständnis der theoretisch-mathematischen Grundlagen von KNN im Kontext des überwachten Lernens.
- Unterscheidung zwischen verschiedenen KNN und Auswahl eines geeigneten Verfahrens in einer Anwendungssituation.
- Selbstständige Implementierung grundlegender Verfahren in einer höheren Programmiersprache, sowie Anwendung der eigenen und anderer Implementierung auf Daten.
- Selbstständige Analyse der Ergebnis der KNN interpretieren, insbesondere Erkennen, wann sie unrealistisch sind.

Inhalt:

Verfahren: Struktur von Optimierungsproblemen, Regression, logistische Regression, biologische neuronal Netze, Modellselektion, universelle Approximationstheorem, Perzeptron, mehrschichtiges Perzeptron, Backpropagation, tiefe neuronale Netze, rekurrente neuronale Netze, Long-Short Term Memory, Hopfield Netze, Boltzmann-Machine

Software: python, numpy, matplotlib, scikit-learn, tensorflow

Lehrformen:

Vorlesung, Hausaufgaben, angeleitete Übungen am Computer

Prüfungsformen:

Klausur, 120 Minuten

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Modulabschlussklausur

Verwendung des Moduls (in anderen Studiengängen): B.Sc. Elektrotechnik und Informationstechnik, M.Sc. Cognitive Science

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Sen Cheng

Sonstige Informationen:**Literatur:**

- **Charu C. Aggarwal: “Neural Networks and Deep Learning”, Springer**
 - Aurélien Géron: “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques for Building Intelligent Systems”, O’Reilly
 - Skript

Mathematics for Modeling and Data Analysis					
Modul-Nr: 310503 (Übung: 310513)	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Mathematics for Modeling and Data Analysis			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppen- größe: 20
Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Basic knowledge of calculus and linear algebra.					
Lernziele (learning outcomes): After the successful completion of this course the students: <ul style="list-style-type: none"> • know the material covered in this course, see Content, • do have an intuitive understanding of the basic concepts and can work with that, • can communicate about all this in English. 					
Inhalt: This course covers mathematical methods that are relevant for modeling and data analysis. Particular emphasis will be put on an intuitive understanding as is required for a creative command of mathematics. The following topics will be covered: Functions, vector spaces, matrices as transformations, systems of linear differential equations, and qualitative analysis of nonlinear differential equations, possibly also Bayesian theory and multiple integrals.					
Lehrformen: Lecture and tutorial					
Prüfungsformen: Oral exam, ca 20 min					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					
Stellenwert der Note für die Endnote: 6 / 175					
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Laurenz Wiskott					
Sonstige Informationen: 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.					

Introduction to Computational Neuroscience					
Modul-Nr: 310506	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Lecture and tutorial			Kontaktzeit: 4 SWS	Selbststudium: 120 h	Gruppen- größe: a) 150 b) 20
Teilnahmevoraussetzungen: Mathematics 1 und 2, Statistics, Programming					
Empfohlene Vorkenntnisse: Neuroscience (not required)					
Lernziele (learning outcomes): After successful completion of this course, students will be able to <ul style="list-style-type: none"> • recognize and describe fundamental models in computational neuroscience • use such models to describe the functioning of the nervous system • choose the appropriate model when addressing a scientific question in neuroscience • analyze the behavior of these models • implement such models in computer programs 					
Inhalt: Computational neuroscience uses quantitative methods to describe what nervous systems do, study how they function, and explain the underlying principles. This class introduces the basics of the mathematical and computational methods used in contemporary neuroscience research. These methods are applied to problems in perception, motor control, learning, and memory.					
Lehrformen: Lecture and tutorial					
Prüfungsformen: Written final exam. 120min.					
Voraussetzungen für die Vergabe von Kreditpunkten: Bestandene Modulabschlussklausur					
Verwendung des Moduls (in anderen Studiengängen): B.Sc. Elektrotechnik und Informationstechnik, M.Sc. Cognitive Science					
Stellenwert der Note für die Endnote: 6 / 175					

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Sen Cheng

Sonstige Informationen:

Im SoSe 2020 zunächst 2+2, in Zukunft ggf. 4+2

Literatur: “Theoretical Neuroscience” by Dayan and Abbott, MIT Press

Programmier- und Simulationstechnik

Game Development					
Modul-Nr: MISSING	Credits: 6 CP	Workload: 180 h	Semester: 3.-6. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Vorlesung + Übung			Kontaktzeit: 2 SWS	Selbststudium: 70 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Grundlegende Programmierkenntnisse; Zugang zu einem Computer mit Internet, auf dem die Unity-Engine läuft (https://unity3d.com/de/get-unity/download)					
Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls <ul style="list-style-type: none"> • verstehen die Teilnehmer Grundlagen der objektorientierten Programmierung mit C#-harp im Rahmen der Unity-Engine, • haben Teilnehmer ein umfassendes Wissen über den Bereich der Spieleentwicklung erworben und kennen moderne Tools sowie aktuelle Methoden der 2D- und 3D-Entwicklung, • können die Teilnehmer praxisnahe Problemstellungen der Softwareentwicklung analysieren und eigenständig lösen, • können die Teilnehmer Projekte im Bereich der Spieleentwicklung definieren und fachgerecht umsetzen. 					
Inhalt: Die Veranstaltung bietet einen umfangreichen Einblick in viele Bereiche der Spieleentwicklung. Dazu gehören: <ul style="list-style-type: none"> • Grundlagenwissen (Spiele-Engines, moderne Softwaretools, Projektmanagement) • C#-Grundlagen (Syntax, Datentypen, Operatoren, Kontrollstrukturen) • Benutzerinteraktion (In-/Output mit Tastatur sowie Controller, User Interfaces) 					

KAPITEL 12. PROGRAMMIER- UND SIMULATIONSTECHNIK

- Gameplay (Bewegen von Spielobjekten, Kamerasteuerung, Game Loop und Framerates)
- Physik (Rigidbody, Collider, Trigger)
- Assets (Import von Bildern, Audio und 3D-Modellen sowie Erstellung von Animationen)
- Grafik (Texturen, Partikeleffekte, Beleuchtung, Post-Processing)
- Leveldesign (Tilemaps, 3D-Umgebungen, Terrains)

Studenten setzen das erlernte Wissen durch die Entwicklung einfacher Computerspiele in der Unity-Engine um. Die erworbenen Fähigkeiten lassen sich jedoch einfach auf andere Software-Frameworks übertragen.

Lehrformen:

Online Vorlesung und begleitende Übung

Prüfungsformen:

Semesterbegleitende Projektarbeiten

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Projektarbeiten; Studienleistung: Bearbeitung wöchentlicher Tests, die sich inhaltlich auf die einzelnen Kursabschnitte beziehen. Für die Zulassung zu den Projektarbeiten müssen die jeweiligen Tests erfolgreich absolviert worden sein.

Verwendung des Moduls (in anderen Studiengängen): Wahlpflichtmodul im Bachelorstudiengang Informatik

Stellenwert der Note für die Endnote: 6 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Tobias Glasmachers
Daniel Vonk

Agent-based modeling in economics and business					
Modul-Nr: 073085	Credits: 5 CP	Workload: 150 h	Semester: 3.-6. Sem.	Turnus: Wintersemester	Dauer: 1 Semester
Lehrveranstaltungen: a) Vorlesung b) Seminar			Kontaktzeit: 2 SWS	Selbststudium: 120 h	Gruppen- größe: 20 Teilnehmer, gemischt: Ang. Informatik und WiWi
Teilnahmevoraussetzungen: Gute Englischkenntnisse, Grundlegende Programmierkenntnisse					
<p>Lernziele (learning outcomes): Das Modul verfolgt das Ziel, Studierende der angewandten Informatik und Studierende der Wirtschaftswissenschaft in die Methode der agenten-basierten Modellierung und Simulationstechniken einzuführen. Hierbei sollen Studierende relevante volkswirtschaftliche Fragestellungen mit quantitativen Ansätzen bearbeiten.</p> <p>Im Rahmen des Moduls erwerben Studierende folgende Kenntnisse:</p> <ol style="list-style-type: none"> 1) Überführung von wissenschaftlichen Fragestellungen aus dem Bereich Volkswirtschaftslehre in agenten-basierte Computersimulationen 2) Praktische Arbeit mit agenten-basierten Modellen, Interpretation von Simulationsergebnissen und mögliche Limitationen des Ansatzes 3) Grundlegendes Wissen, um eigene Modelle zu implementieren und Simulationen selbstständig durchzuführen 4) Die Programmiersprache NetLogo 					
<p>Inhalt: In der Vorlesung wird die Methode der agentenbasierten Computersimulation vorgestellt und gezeigt, wie sie zur Analyse komplexer ökonomischer Systeme angewendet werden kann. Die Methode der agenten-basierten Computersimulationen gewinnt in der Forschung sowie in der Praxis weiter an Relevanz. Dazu werden Beispiele aus der betriebswirtschaftlichen und der volkswirtschaftlichen Forschungsliteratur präsentiert. Während der Seminartermine werden verschiedene Modelle vorgestellt, die dann als Grundlage für Gruppendiskussionen und eigene Programmieraufgaben dienen. Hier steht die praktische Anwendung im Vordergrund.</p>					
Lehrformen: Vorlesung (+ Seminar)					
Prüfungsformen: Semesterbegleitend; Hausarbeit in Form eines Lab-Reports Vortrag im Seminarkolloquium					
Voraussetzungen für die Vergabe von Kreditpunkten:					

KAPITEL 12. PROGRAMMIER- UND SIMULATIONSTECHNIK

Bestehen der folgenden Leistungen:

- 1) Studienleistung (unbenotet)
- 2) Hausarbeit und Vortrag (benotet)

Stellenwert der Note für die Endnote: 5 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Michael Roos

Sonstige Informationen:

Nicht im WS 21/22! 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.

Kein Angebot im WS 20/21

Literatur: Die Leseliste und geeignete Lehrbücher werden zu Beginn der Veranstaltung bekanntgeben.

Vertiefungsseminar

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

KAPITEL 13. VERTIEFUNGSSEMINAR

- Analytisches und logisches Denken
- Fächerübergreifendes Denken und Arbeiten
- Literaturrecherche und Dokumentation
- Projekt- und Zeitmanagement
- Präsentation wissenschaftlicher Ergebnisse
- Selbständiges Lernen und Arbeiten

Inhalt:

Gegenstand des Moduls sind unterschiedliche Themen aus den Bereichen Computerlinguistik und Sprachtechnologie, sprich deren spezifische Untersuchungsgegenstände und Methoden.

Seminare: Typische Themen einschlägiger und in unterschiedlichem Turnus angebotener Seminare sind u. a.

- Korpuslinguistik
- Data mining, Explorative Datenanalyse (mit R)
- Klassifikationsalgorithmen
- Natürlichsprachliche Systeme, Automatische Sprachverarbeitung, Automatische Lexikonakquise
- Automatische Textzusammenfassung, Textgenerierung
- Semantische Verarbeitung in der Computerlinguistik
- Morphologische und syntaktische Analyse in der Computerlinguistik

Lehrformen:

Vertiefungsseminare

Prüfungsformen:

Semesterbegleitend; Seminar-Arbeit

Voraussetzungen für die Vergabe von Kreditpunkten:

Bestandene Seminar-Arbeit

Stellenwert der Note für die Endnote: 3 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Johanna Marie Poppek, M. A.

Roussel, Simonjetz, Klabunde

Sonstige Informationen:

Angebot siehe VVZ

Die Teilnahme ist nur nach vorheriger Anmeldung in CampusOffice möglich. Diese An-

KAPITEL 13. VERTIEFUNGSEMINAR

meldung 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.

Literatur: Wird im Kurs bekanntgegeben.

Seminar Ingenieurinformatik					
Modul-Nr: 125017	Credits: 3 CP	Workload: 90 h	Semester: 4.-6.. Sem.	Turnus: Winter- und Sommersemes- ter	Dauer: 1 Semester
Lehrveranstaltungen: Seminar Ingenieurinformatik			Kontaktzeit: 2 SWS	Selbststudium: 60 h	Gruppen- größe: beliebig
<p>Teilnahmevoraussetzungen: Keine Empfohlene Vorkenntnisse: Mindestens ein Vertiefungsmodul der Ingenieurinformatik</p>					
<p>Lernziele (learning outcomes): 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"> • kennen Studierende die methodischen Ansätze und Techniken, um sich in eine Thematik gründlich einzuarbeiten. Dabei sollen die gewonnenen Erkenntnisse in verständlicher Form weitergegeben werden; • verstehen Studierende durch Bearbeitung einer Seminararbeit, wie die erarbeiteten Fakten, Informationen und Zusammenhänge in Rahmen einer Präsentation klar vermittelt werden können. <p>Vermittelte Kompetenzen: Kernkompetenzen:</p> <ul style="list-style-type: none"> • Fächerübergreifendes Denken und Arbeiten • Kritikfähigkeit • Literaturrecherche und Dokumentation • Projekt- und Zeitmanagement • Präsentation wissenschaftlicher Ergebnisse • Rhetorik und sprachliche Kompetenz • Selbständiges Lernen und Arbeiten 					

KAPITEL 13. VERTIEFUNGSSEMINAR

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 / 175

Modulbeauftragte/r und hauptamtlich Lehrende:

Prof. Dr. Markus König
Dr. K. Lehner

Seminar Kryptologie und Theoretische Informatik					
Modul-Nr: 150537: Se- minar zur Kryptographie	Credits: 3 CP	Workload: 90 h	Semester: BA: 4.-6.; MA: 2.-3.. Sem.	Turnus: Unregelmäßig	Dauer: 1 Semester
Lehrveranstaltungen: Vertiefungsseminare			Kontaktzeit: 2 SWS	Selbststudium: 60 h	Gruppen- größe: beliebig
Teilnahmevoraussetzungen: Keine					
Empfohlene Vorkenntnisse: Kryptographie, Kryptanalyse, Algorithmik					
Lernziele (learning outcomes): Es werden aktuelle Forschungsarbeiten der Kryptologie und Algorithmik vorgestellt					
Vermittelte Kompetenzen: Kernkompetenzen:					
<ul style="list-style-type: none"> • Fächerübergreifendes Denken und Arbeiten • Kritikfähigkeit • Literaturrecherche und Dokumentation • Projekt- und Zeitmanagement • Präsentation wissenschaftlicher Ergebnisse • Rhetorik und sprachliche Kompetenz • Selbständiges Lernen und Arbeiten 					
Inhalt: Inhalt entsprechend des gewählten Seminars.					
Lehrformen: Vertiefungsseminare					
Prüfungsformen: Semesterbegleitend; Seminar-Arbeit					
Voraussetzungen für die Vergabe von Kreditpunkten: Erfolgreicher Seminarvortrag					
Stellenwert der Note für die Endnote: 3 / 175					
Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Alexander May					

KAPITEL 13. VERTIEFUNGSSSEMINAR

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

Sonstige Informationen:

Angebot siehe VVZ

Bachelor-Seminar Maschinelles Lernen					
Modul-Nr: 310528	Credits: 3 CP	Workload: 90 h	Semester: 4.-6.. Sem.	Turnus: Sommersemester	Dauer: 1 Semester
Lehrveranstaltungen: Seminar (2 SWS)			Kontaktzeit: 2 SWS	Selbststudium: 60 h	Gruppen- größe: max. 20 Teil- nehmer
Teilnahmevoraussetzungen: Keine					
<p>Lernziele (learning outcomes): Nach dem erfolgreichen Abschluss des Moduls</p> <ul style="list-style-type: none"> • kennen die Teilnehmer Algorithmen und Anwendungen des maschinellen Lernens • können die Teilnehmer Lehrinhalte und Forschungsergebnisse eigenständig in einem didaktisch wohl aufbereiteten Vortrag vermitteln, • können die Teilnehmer konstruktives Feedback formulieren und entgegennehmen. 					
<p>Inhalt: Das Seminar vertieft diverse Inhalt aus dem breiten Bereich des maschinellen Lernens. Weiterhin werden guter Vortragsstil und das Üben sowie die Entgegennahme konstruktiver Kritik geübt.</p>					
<p>Lehrformen: Seminarvorträge und Feedbackrunden</p>					
<p>Prüfungsformen: Seminarvortrag</p>					
<p>Voraussetzungen für die Vergabe von Kreditpunkten: erfolgreiche Präsentation des Seminarvortrags</p>					
<p>Stellenwert der Note für die Endnote: 3 / 175</p>					
<p>Modulbeauftragte/r und hauptamtlich Lehrende: Prof. Dr. Tobias Glasmachers</p>					
<p>Sonstige Informationen: Um die Lernziele zu erreichen, besteht in diesem Seminar Anwesenheitspflicht an mindestens 9 von 10 Terminen. Mehrfaches Fehlen muss durch ein ärztliches Attest entschuldigt werden. Die Anwesenheit beim ersten Termin ist obligatorisch, da zu diesem Termin die Themen verteilt werden. Das Seminar gilt als nicht bestanden, wenn an mehr als einem Termin unentschuldigt gefehlt wurde.</p>					