



Modulhandbuch

Master of Science (M.Sc.)

Computer Science

Kohorte: Wintersemester 2021

Stand: 31. Mai 2024

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Studiengangsbeschreibung	3
Fachmodule der Kernqualifikation	5
Modul M0523: Betrieb & Management	5
Modul M0524: Nichttechnische Angebote im Master	6
Modul M1563: Forschungsprojekt Informatik	8
Fachmodule der Vertiefung I. Computer- und Software-Engineering	9
Modul M0753: Software Verification	9
Modul M0942: Software Security	11
Modul M1694: Security of Cyber-Physical Systems	13
Modul M0926: Verteilte Algorithmen	15
Modul M1400: Entwurf von Dependable Systems	16
Modul M1685: Ausgewählte Aspekte der Informatik	18
Modul M1682: Secure Software Engineering	19
Modul M1812: Constraint Satisfaction Problems	20
Modul M1886: GPU Arcuitectures and Programming	21
Modul M1397: Modellprüfung - Beweiser und Algorithmen	22
Modul M1301: Software Testing	24
Modul M1427: Algorithmische Spieltheorie	26
Modul M1248: Compiler für Eingebettete Systeme	28
Modul M0556: Computer Graphics	30
Modul M1741: Betriebssystembau	32
Modul M1810: Autonomous Cyber-Physical Systems	34
Modul M1774: Advanced Internet Computing	35
Modul M0924: Software für Eingebettete Systeme	37
Modul M0910: Fortgeschrittener Entwurf von Chip-Systemen (Praktikum)	39
Modul M0839: Traffic Engineering	40
Modul M1742: Betriebssystemtechnik	42
Modul M1780: Massively Parallel Systems: Architecture and Programming	44
Fachmodule der Vertiefung II. Intelligenz-Engineering	46
Modul M0633: Industrial Process Automation	46
Modul M0550: Digital Image Analysis	48
Modul M0629: Intelligent Autonomous Agents and Cognitive Robotics	50
Modul M0630: Robotics and Navigation in Medicine	52
Modul M1702: Process Imaging	54
Modul M0627: Machine Learning and Data Mining	56
Modul M1302: Angewandte Humanoide Robotik	58
Modul M1249: Medizinische Bildgebung	59
Modul M0623: Intelligent Systems in Medicine	61
Fachmodule der Vertiefung III. Mathematik	63
Modul M0667: Algorithmische Algebra	63
Modul M1428: Lineare und Nichtlineare Optimierung	65
Modul M0716: Hierarchische Algorithmen	67
Modul M1405: Randomisierte Algorithmen und Zufällige Graphen	69
Modul M0714: Numerik gewöhnlicher Differentialgleichungen	71
Modul M1668: Probability Theory	73
Modul M0711: Numerische Mathematik II	75
Modul M0881: Mathematische Bildverarbeitung	77
Modul M1552: Fortgeschrittenes maschinelles Lernen	79
Modul M1020: Numerik partieller Differentialgleichungen	81
Fachmodule der Vertiefung IV. Fachspezifische Fokussierung	83
Modul M1565: Technischer Ergänzungskurs I für CSMS	83
Modul M1566: Technischer Ergänzungskurs II für CSMS	84
Modul M1564: Hauptseminare Informatik und Kommunikationstechnik	85
Thesis	86
Modul M-002: Masterarbeit	86

Studiengangsbeschreibung

Inhalt

Die Informatik hat sich zu einer Triebfeder des technologischen Fortschrittes entwickelt, weil alle Berufszweige mit Informatikaspekten durchdrungen sind und immer noch zusätzliche Anwendungsfelder in der Informations- und Kommunikationstechnik erschlossen werden. Der Masterstudiengang Computer Science trägt dieser Entwicklung Rechnung. Es werden die kreativen und konstruktiven Fähigkeiten zur Neu- und Weiterentwicklung von IT-Systemen sowie die ökonomischen und Management-Kompetenzen zur Planung und Durchführung umfangreicher IT-Projekte gefördert. Da IT-Systeme größtenteils durch das Zusammenwirken von Mensch, Technik, Unternehmen und Gesellschaft entstehen, sind Anwendungswissen, Sozialkompetenz und ethisches Verantwortungsbewusstsein unverzichtbare Bestandteile der Ausbildung. Des Weiteren bereitet das Masterstudium auch auf eine Promotion in Informatik vor.

Im Masterstudiengang Computer Science wird ein breites, fundiertes und vertieftes Grundlagenwissen in den Bereichen mathematische Modellbildung in der Informatik, Softwaretechnik, Hardwareentwurf und Intelligente Systeme vermittelt. Zudem werden weitergehende Kenntnisse in Betriebswirtschaftslehre und Management sowie nichttechnischen Fächern erlangt, um die Kompetenzen für die Bewältigung umfangreicher IT-Projekte zu erhöhen. Das Masterprogramm bereitet auf Kompetenzebene sowohl auf praktische Berufsfelder als auch auf die Forschung in Informatik vor.

Berufliche Perspektiven

Der Masterstudiengang Computer Science bereitet die Absolventen und Absolventinnen sowohl auf eine berufliche Tätigkeit im IT-Sektor als auch auf eine Promotion in Informatik vor.

Der Studiengang bildet Informatikinnen und Informatiker aus, die auf dem deutschen und internationalen Arbeitsmarkt unabhängig von Konjunkturbewegungen sehr gute Beschäftigungsmöglichkeiten vorfinden. Absolventen und Absolventinnen werden nicht nur als Systementwickler in der IT-Branche oder in den Entwicklungsabteilungen des Maschinenbaus und der Automobilindustrie tätig sein, sondern auch in der Medienindustrie als Entwickler von autonom agierenden Systemen oder Computerspielen.

Lernziele

Das Masterstudium Computer Science soll die Studierenden sowohl auf eine gehobene berufliche Tätigkeit im IT-Bereich als auch auf die Forschung in Informatik vorbereiten. Die dazu notwendigen methodischen Kompetenzen werden im Rahmen des Studiums erworben. Die Lernziele sind im Folgenden eingeteilt in die Kategorien Wissen, Fertigkeiten, Sozialkompetenz und Selbstständigkeit.

Wissen

Wissen konstituiert sich aus Fakten, Grundsätzen und Theorien und wird im Masterstudiengang Computer Science auf folgenden Gebieten erworben:

1. Die Absolventen und Absolventinnen kennen in detaillierter Weise aktuelle Methoden und Verfahren zur mathematischen Modellbildung in der Informatik, wie etwa Agentensysteme, Bayessche Netze, dynamische Systeme, dynamische Programme, Entscheidungsbäume, lineare und nichtlineare (ganzzahlige) Programme, künstliche neuronale Netze sowie Fuzzylogik. Sie können diese Modelle detailliert beschreiben und verschiedene Repräsentationsformen desselben Modells vergleichen.
2. Die Absolventen und Absolventinnen kennen Punkt für Punkt weitergehende Methoden und Verfahren zur Lösung oder Approximation von algorithmischen Entscheidungs- und Optimierungsaufgaben, wie etwa Algorithmen in Netzwerken, Inferenz- und Lernverfahren für probabilistische Modelle, dynamische Programmierung, lineare und nichtlineare (ganzzahlige) Programmierung, Verfahren zur Lösung von Ausgleichsproblemen, Eigenwertproblemen und nichtlinearen Nullstellenproblemen, Klassifikation mit künstlichen neuronalen Netzen, Bildung von Gruppierungen (Clustering) sowie Numerik und Algorithmik für Hochleistungsrechner.
3. Die Absolventen und Absolventinnen kennen im Detail weiterführende Methoden und Verfahren der Softwaretechnik, insbesondere Methoden zur Analyse und Verifikation von Software, die Konzipierung von Web- und Cloud-Diensten, den Entwurf von Spiele-Software und Verfahren des verteilten Rechnens.
4. Die Absolventen und Absolventinnen verstehen in Einzelheiten, wie Instanzen von Hardware-Modellen durch Verhaltens- und Strukturbeschreibungen spezifiziert werden und können die Einbettung von Strukturmodellen in einen technischen Rahmen unter Einbeziehung von Betriebssystem- und Netzwerkkomponenten beschreiben. Hierbei sind sie in der Lage auf Kenntnisse in den Bereichen Codierung und Decodierung von Daten, eingebettete Systeme, Kommunikationsnetze, Netzwerksicherheit, Sensornetze und Warteschlangenmodelle zurückzugreifen.
5. Die Absolventen und Absolventinnen sind vertraut mit den Grundzügen komplexer informations- und kommunikationstechnischer Systeme, so genannter cyber-physischer Systeme. Dies beinhaltet relevante Steuerungsarchitekturen, Interaktionsmechanismen, Sensorik und Aktorik und die Gewinnung und Verarbeitung von Wissen und Erkenntnissen aus dem System heraus.
6. Die Absolventen und Absolventinnen kennen im Einzelnen eine ganze Reihe von Anwendungsfällen valider mathematischer Modelle in der Informatik, wie etwa Algorithmen in Netzwerken, Bayessche Netze für die Analyse von Markov-Ketten in diversen Anwendungen, Medizintechnik, Robotik und Kameraführung sowie Methoden des Operations-Research für betriebswirtschaftliche und technische Planungsprobleme.

Fertigkeiten

Die Fähigkeit, erlerntes Wissen anzuwenden, um spezifische Probleme zu lösen, wird im Studiengang Computer Science auf vielfältige Weise unterstützt:

1. Die Absolventinnen und Absolventen sind in der Lage, Instanzen formaler Modelle in der Informatik anhand weitergehender Modellierungsansätze zu entwickeln, ihre Berechenbarkeit und Komplexität direkt oder durch Reduktion zu ermitteln und sie vermöge geeigneter Programmierwerkzeuge in einem technischen Rahmen umzusetzen.
2. Die Absolventinnen und Absolventen sind imstande, Instanzen von algorithmischen Entscheidungs- und Optimierungsproblemen unter Verwendung weiterführender Verfahren und unter Einsatz einschlägiger Software-Werkzeuge optimal bzw. näherungsweise zu lösen und die Lösungen zu evaluieren.
3. Die Absolventen und Absolventinnen können komplexe Softwaresysteme, wie etwa Web- und Cloud-Dienste, Spiele-Software und nebenläufige Systeme, entwickeln und diese analysieren und verifizieren.
4. Die Absolventen und Absolventinnen sind in der Lage, Strukturbeschreibungen von komplexen Hardware-Bausteinen, wie etwa komplette CPUs, eingebettete Systeme, Coprozessoren oder Mikroprozessorsysteme, unter Verwendung spezifischer Entwicklungswerkzeuge zu konzipieren und zu evaluieren.
5. Die Absolventen und Absolventinnen können Komponenten von cyber-physischen Systemen unter Einsatz spezifischer Methoden und Verfahren entwickeln und in größere Systeme integrieren und testen.
6. Die Absolventen und Absolventinnen sind im Stande, weiterführende valide mathematische Modelle aus der Informatik unter Verwendung einer geeigneten Programmier- und Testumgebung technisch umzusetzen und zu validieren.

Erwerb von Sozialkompetenz

Sozialkompetenz umfasst die individuelle Fähigkeit und den Willen, zielorientiert mit anderen zusammen zu arbeiten, die Interessen der anderen zu erfassen, sich zu verständigen und die Arbeits- und Lebenswelt mitzugestalten.

1. Die Absolventen und Absolventinnen können Teamsitzungen und Gruppenprojektarbeiten zu einem Thema aus der Informatik leiten und durchführen.
2. Die Absolventen und Absolventinnen sind in der Lage, Lösungen von Aufgabenstellungen aus der Informatik vor einer Hörerschaft mit Fachvertretern zu vertreten.

Kompetenz zum selbstständigen Arbeiten

Personale Kompetenzen umfassen neben der Kompetenz zum selbstständigen Handeln auch die System- und Lösungskompetenz, allgemeine Problemstellungen auf spezifische Teilprobleme abzubilden sowie die Auswahl und das Beherrschen geeigneter Methoden und Verfahren zur Problemlösung.

1. Die Absolventen und Absolventinnen können eigenständig ein Thema aus der Informatik erschließen und die Ergebnisse im Rahmen eines Vortrages mit fortgeschrittenen Präsentationstechniken oder anhand einer fundierten Abhandlung gemäß den Grundsätzen guter wissenschaftlicher Praxis darstellen.
2. Die Absolventen und Absolventinnen sind im Stande, zeitlich begrenzte und ressourcenbeschränkte Forschungsaufgaben unter Reflexion des im Studium Erlernten eigenverantwortlich durchzuführen.
- 3.

Studiengangsstruktur

Das Curriculum des Masterstudienganges Computer Science ist wie folgt gegliedert:

- Kernqualifikation:
 - Forschungsprojekt & Seminar (18 LP), bestehend aus Projektarbeit (15 LP) und Hauptseminar (3 LP)
 - Betrieb & Management (6 LP)
 - Nichttechnische Ergänzungsfächer (6 LP)
- Vertiefung:
 - Vertiefungsrichtungen: Computer- und Software-Engineering, Intelligenz-Engineering
 - Wahl von 10 Modulen aus der gewählten Vertiefung (60 LP).
 - In jede Vertiefungen sind zwei Technische Ergänzungsfächer integriert, die aus dem Gesamtbereich der technischen Fächer der TU gewählt werden können.
- Masterarbeit: 30 LP, 4. Semester

Dadurch ergibt sich ein Gesamtaufwand von 120 LP. Der Studienplan enthält ein Mobilitätsfenster, so dass Studierende das dritte Semester im Ausland absolvieren können.

Fachmodule der Kernqualifikation

Die Kernqualifikation setzt sich aus mehreren Komponenten zusammen:

- einem Studium-Generale-Anteil von 12 LP, bestehend aus betriebswirtschaftlichen und nichttechnischen Vertiefungsveranstaltungen,
- einem Informatik-Forschungsprojekt von 10 LP, das einer Projektarbeit mit einem forschungsbezogenen Thema entspricht,
- einem Hauptseminar von 2 LP, in dem sich die Studierenden mit einem Thema aus der Informatik intensiv beschäftigen sollen, und
- einer Masterarbeit von 30 LP, im Rahmen derer eine wissenschaftliche Arbeit selbständig, aber unter Betreuung zu verfassen ist.

Näheres zum Informatik-Forschungsprojekt und zur Masterarbeit regelt die FSPO des Studienganges.

Modul M0523: Betrieb & Management	
Modulverantwortlicher	Prof. Matthias Meyer
Zulassungsvoraussetzungen	Keine
Empfohlene Vorkenntnisse	Keine
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht
Fachkompetenz <i>Wissen</i>	• Die Studierenden sind in der Lage, ausgewählte betriebswirtschaftliche Spezialgebiete innerhalb der Betriebswirtschaftslehre zu verorten. • Die Studierenden können in ausgewählten betriebswirtschaftlichen Teilbereichen grundlegende Theorien, Kategorien und Modelle erklären. • Die Studierenden können technisches und betriebswirtschaftliches Wissen miteinander in Beziehung setzen.
<i>Fertigkeiten</i>	• Die Studierenden können in ausgewählten betriebswirtschaftlichen Teilbereichen grundlegende Methoden anwenden. • Die Studierenden können für praktische Fragestellungen in betriebswirtschaftlichen Teilbereichen Entscheidungsvorschläge begründen.
Personale Kompetenzen <i>Sozialkompetenz</i>	• Die Studierenden sind in der Lage, in interdisziplinären Kleingruppen zu kommunizieren und gemeinsam Lösungen für komplexe Problemstellungen zu erarbeiten.
<i>Selbstständigkeit</i>	• Die Studierenden sind in der Lage, sich notwendiges Wissen durch Recherchen und Aufbereitungen von Material selbstständig zu erschließen.
Arbeitsaufwand in Stunden	Abhängig von der Wahl der Lehrveranstaltungen
Leistungspunkte	6

Lehrveranstaltungen
Die Informationen zu den Lehrveranstaltungen entnehmen Sie dem separat veröffentlichten Modulhandbuch des Moduls.

Modul M0524: Nichttechnische Angebote im Master	
Modulverantwortlicher	Dagmar Richter
Zulassungsvoraussetzungen	Keine
Empfohlene Vorkenntnisse	Keine
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht
Fachkompetenz <i>Wissen</i>	Die Nichttechnischen Angebote (NTA) vermittelt die in Hinblick auf das Ausbildungsprofil der TUHH nötigen Kompetenzen, die ingenieurwissenschaftliche Fachlehre fördern aber nicht abschließend behandeln kann: Eigenverantwortlichkeit, Selbstführung, Zusammenarbeit und fachliche wie personale Leitungsbefähigung der zukünftigen Ingenieurinnen und Ingenieure. Er setzt diese Ausbildungsziele in seiner Lehrarchitektur, den Lehr-Lern-Arrangements, den Lehrbereichen und durch Lehrangebote um, in denen sich Studierende wahlweise für spezifische Kompetenzen und ein Kompetenzniveau auf Bachelor- oder Masterebene qualifizieren können. Die Lehrangebote sind jeweils in einem Modulkatalog Nichttechnische Ergänzungskurse zusammengefasst. Die Lehrarchitektur besteht aus einem studiengangübergreifenden Pflichtstudienangebot. Durch dieses zentral konzipierte Lehrangebot wird die Profilierung der TUHH Ausbildung auch im nichttechnischen Bereich gewährleistet. Die Lernarchitektur erfordert und übt eigenverantwortliche Bildungsplanung in Hinblick auf den individuellen Kompetenzaufbau ein und stellt dazu Orientierungswissen zu thematischen Schwerpunkten von Veranstaltungen bereit. Das über den gesamten Studienverlauf begleitend studierbare Angebot kann ggf. in ein-zwei Semestern studiert werden. Angesichts der bekannten, individuellen Anpassungsprobleme beim Übergang von Schule zu Hochschule in den ersten Semestern und um individuell geplante Auslandssemester zu fördern, wird jedoch von einer Studienfixierung in konkreten Fachsemestern abgesehen. Die Lehr-Lern-Arrangements sehen für Studierende - nach B.Sc. und M.Sc. getrennt - ein semester- und fachübergreifendes voneinander Lernen vor. Der Umgang mit Interdisziplinarität und einer Vielfalt von Lernständen in Veranstaltungen wird eingeübt - und in spezifischen Veranstaltungen gezielt gefördert. Die Lehrbereiche basieren auf Forschungsergebnissen aus den wissenschaftlichen Disziplinen Kulturwissenschaften, Gesellschaftswissenschaften, Kunst, Geschichtswissenschaften, Kommunikationswissenschaften, Migrationswissenschaften, Nachhaltigkeitsforschung und aus der Fachdidaktik der Ingenieurwissenschaften. Über alle Studiengänge hinweg besteht im Bachelorbereich zusätzlich ab Wintersemester 2014/15 das Angebot, gezielt Betriebswirtschaftliches und Gründungswissen aufzubauen. Das Lehrangebot wird durch soft skill und Fremdsprachkurse ergänzt. Hier werden insbesondere kommunikative Kompetenzen z.B. für Outgoing Engineers gezielt gefördert. Das Kompetenzniveau der Veranstaltungen in den Modulen der nichttechnischen Ergänzungskurse unterscheidet sich in Hinblick auf das zugrunde gelegte Ausbildungsziel: Diese Unterschiede spiegeln sich in den verwendeten Praxisbeispielen, in den - auf unterschiedliche berufliche Anwendungskontexte verweisende - Inhalten und im für M.Sc. stärker wissenschaftlich-theoretischen Abstraktionsniveau. Die Soft skills für Bachelor- und für Masterabsolventinnen/ Absolventen unterscheidet sich an Hand der im Berufsleben unterschiedlichen Positionen im Team und bei der Anleitung von Gruppen. Fachkompetenz (Wissen) Die Studierenden können • ausgewählte Spezialgebiete des jeweiligen nichttechnischen Bereiches erläutern, • in den im Lehrbereich vertretenen Disziplinen grundlegende Theorien, Kategorien, Begrifflichkeiten, Modelle, Konzepte oder künstlerischen Techniken skizzieren, • diese fremden Fachdisziplinen systematisch auf die eigene Disziplin beziehen, d.h. sowohl abgrenzen als auch Anschlüsse benennen, • in Grundzügen skizzieren, inwiefern wissenschaftliche Disziplinen, Paradigmen, Modelle, Instrumente, Verfahrensweisen und Repräsentationsformen der Fachwissenschaften einer individuellen und soziokulturellen Interpretation und Historizität unterliegen, • können Gegenstandsangemessen in einer Fremdsprache kommunizieren (sofern dies der gewählte Schwerpunkt im NTW-Bereich ist).
<i>Fertigkeiten</i>	Die Studierenden können in ausgewählten Teilbereichen • grundlegende und teils auch spezielle Methoden der genannten Wissenschaftsdisziplinen anwenden. • technische Phänomene, Modelle, Theorien usw. aus der Perspektive einer anderen, oben erwähnten Fachdisziplin befragen. • einfache und teils auch fortgeschrittene Problemstellungen aus den behandelten Wissenschaftsdisziplinen erfolgreich bearbeiten, • bei praktischen Fragestellungen in Kontexten, die den technischen Sach- und Fachbezug übersteigen, ihre Entscheidungen zu Organisations- und Anwendungsformen der Technik begründen.

Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>	Die Studierenden sind fähig , • in unterschiedlichem Ausmaß kooperativ zu lernen • eigene Aufgabenstellungen in den o.g. Bereichen in adressatengerechter Weise in einer Partner- oder Gruppensituation zu präsentieren und zu analysieren, • nichttechnische Fragestellungen einer Zuhörerschaft mit technischem Hintergrund verständlich darzustellen • sich landessprachlich kompetent, kulturell angemessen und geschlechtersensibel auszudrücken (sofern dies der gewählte Schwerpunkt im NTW-Bereich ist) Die Studierenden sind in ausgewählten Bereichen in der Lage, • die eigene Profession und Professionalität im Kontext der lebensweltlichen Anwendungsgebiete zu reflektieren, • sich selbst und die eigenen Lernprozesse zu organisieren, • Fragestellungen vor einem breiten Bildungshorizont zu reflektieren und verantwortlich zu entscheiden, • sich in Bezug auf ein nichttechnisches Sachthema mündlich oder schriftlich kompetent auszudrücken. • sich als unternehmerisches Subjekt zu organisieren, (sofern dies ein gewählter Schwerpunkt im NTW-Bereich ist).
Arbeitsaufwand in Stunden	Abhängig von der Wahl der Lehrveranstaltungen
Leistungspunkte	6

Lehrveranstaltungen	
Die Informationen zu den Lehrveranstaltungen entnehmen Sie dem separat veröffentlichten Modulhandbuch des Moduls.	

Modul M1563: Forschungsprojekt Informatik			
Lehrveranstaltungen			
Titel Forschungsprojekt Informatik (L2353)	Typ Projektierungskurs	SWS 8	LP 12
Modulverantwortlicher	Dozenten des SD E		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Wissen und Fertigkeiten aus den Master-Veranstaltungen im 1. und 2. Semester.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Die Studierenden wissen, wie man sich ein Teilgebiet der Informatik selbständig erschließt und selbständig vertiefte Kenntnisse in diesem Gebiet erwirbt. <i>Fertigkeiten</i> Die Studierenden sind in der Lage, wissenschaftliche Problemstellungen zu beschreiben und die selbständig erarbeiteten Lösungswege in einer gut strukturierten Form umzusetzen. Personale Kompetenzen <i>Sozialkompetenz</i> Studierenden erläutern die in einem wissenschaftlichen Aufsatz geschilderten Probleme und die im Aufsatz entwickelten Lösungen aus einem Fachgebiet der Informatik, bewerten die vorgeschlagenen Lösungen in einem Vortrag und reagieren auf wissenschaftliche Nachfragen, Ergänzungen und Kommentare. <i>Selbstständigkeit</i> Die Studierenden können ein Teilgebiet der Informatik zeitgerecht in einer Präsentation vorstellen. Sie können aktiv die Präsentationen anderer Studierender verfolgen, so dass evtl. ein interaktiver Diskurs über ein wissenschaftliches Thema entsteht.		
Arbeitsaufwand in Stunden	Eigenstudium 248, Präsenzstudium 112		
Leistungspunkte	12		
Studienleistung	Keine		
Prüfung	Studienarbeit		
Prüfungsdauer und -umfang	Vortrag		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht		

Lehrveranstaltung L2353: Forschungsprojekt Informatik	
Typ	Projektierungskurs
SWS	8
LP	12
Arbeitsaufwand in Stunden	Eigenstudium 248, Präsenzstudium 112
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	
Literatur	

Fachmodule der Vertiefung I. Computer- und Software-Engineering

Modul M0753: Software Verification

Lehrveranstaltungen

Titel	Typ	SWS	LP
Softwareverifikation (L0629)	Vorlesung	2	3
Softwareverifikation (L0630)	Gruppenübung	2	3

Modulverantwortlicher	Prof. Sibylle Schupp
------------------------------	----------------------

Zulassungsvoraussetzungen	None
----------------------------------	------

Empfohlene Vorkenntnisse	- Automata theory and formal languages - Computational logic - Object-oriented programming, algorithms, and data structures - Functional programming or procedural programming - Concurrency
---------------------------------	--

Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht
---	---

Fachkompetenz <i>Wissen</i>	Students apply the major verification techniques in model checking and deductive verification. They explain in formal terms syntax and semantics of the underlying logics, and assess the expressivity of different logics as well as their limitations. They classify formal properties of software systems. They find flaws in formal arguments, arising from modeling artifacts or underspecification.
<i>Fertigkeiten</i>	Students formulate provable properties of a software system in a formal language. They develop logic-based models that properly abstract from the software under verification and, where necessary, adapt model or property. They construct proofs and property checks by hand or using tools for model checking or deductive verification, and reflect on the scope of the results. Presented with a verification problem in natural language, they select the appropriate verification technique and justify their choice.
Personale Kompetenzen <i>Sozialkompetenz</i>	Students discuss relevant topics in class. They defend their solutions orally. They communicate in English.
<i>Selbstständigkeit</i>	Using accompanying on-line material for self study, students can assess their level of knowledge continuously and adjust it appropriately. Working on exercise problems, they receive additional feedback. Within limits, they can set their own learning goals. Upon successful completion, students can identify and precisely formulate new problems in academic or applied research in the field of software verification. Within this field, they can conduct independent studies to acquire the necessary competencies and compile their findings in academic reports. They can devise plans to arrive at new solutions or assess existing ones.

Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56
----------------------------------	-------------------------------------

Leistungspunkte	6
------------------------	---

Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Ja	15 %	Übungsaufgaben

Prüfung	Klausur
----------------	---------

Prüfungsdauer und -umfang	90 min
----------------------------------	--------

Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme: Pflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Informationstechnologie: Wahlpflicht
---	---

Lehrveranstaltung L0629: Software Verification	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	WiSe
Inhalt	• Syntax and semantics of logic-based systems • Deductive verification ◦ Specification ◦ Proof obligations ◦ Program properties ◦ Automated vs. interactive theorem proving • Model checking ◦ Foundations ◦ Property languages ◦ Tool support • Timed automata • Recent developments of verification techniques and applications
Literatur	• C. Baier and J-P. Katoen, Principles of Model Checking, MIT Press 2007. • M. Huth and M. Bryan, Logic in Computer Science. Modelling and Reasoning about Systems, 2nd Edition, 2004. • Selected Research Papers

Lehrveranstaltung L0630: Software Verification	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0942: Software Security				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Software-Sicherheit (L1103)		Vorlesung	2	3
Software-Sicherheit (L1104)		Gruppenübung	2	3
Modulverantwortlicher	Prof. Riccardo Scandariato			
Zulassungsvoraussetzungen	None			
Empfohlene Vorkenntnisse	Familiarity with C/C++, web programming			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz				
<i>Wissen</i>	Students can - name the main causes for security vulnerabilities in software - explain current methods for identifying and avoiding security vulnerabilities - explain the fundamental concepts of code-based access control			
<i>Fertigkeiten</i>	Students are capable of - performing a software vulnerability analysis - developing secure code			
Personale Kompetenzen				
<i>Sozialkompetenz</i>	None			
<i>Selbstständigkeit</i>	Students are capable of acquiring knowledge independently from professional publications, technical standards, and other sources, and are capable of applying newly acquired knowledge to new problems.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Keine			
Prüfung	Klausur			
Prüfungsdauer und -umfang	120 Minuten			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme: Wahlpflicht			

Lehrveranstaltung L1103: Software Security	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Riccardo Scandariato
Sprachen	EN
Zeitraum	WiSe
Inhalt	- Reliability and Software Security - Attacks exploiting character and integer representations - Buffer overruns - Vulnerabilities in memory management: double free attacks - Race conditions - SQL injection - Cross-site scripting and cross-site request forgery - Testing for security; taint analysis - Type safe languages - Development processes for secure software - Code-based access control
Literatur	M. Howard, D. LeBlanc: Writing Secure Code, 2nd edition, Microsoft Press (2002) G. Hoglund, G. McGraw: Exploiting Software, Addison-Wesley (2004) L. Gong, G. Ellison, M. Dageforde: Inside Java 2 Platform Security, 2nd edition, Addison-Wesley (2003) B. LaMacchia, S. Lange, M. Lyons, R. Martin, K. T. Price: .NET Framework Security, Addison-Wesley Professional (2002) D. Gollmann: Computer Security, 3rd edition (2011)

Lehrveranstaltung L1104: Software Security	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Riccardo Scandariato
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1694: Security of Cyber-Physical Systems			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Sicherheit von Cyber-physischen Systemen (L2691)	Vorlesung	2	3
Sicherheit von Cyber-physischen Systemen (L2692)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Sibylle Fröschle		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	IT security, programming skills, statistics		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	The students know and can explain - the threats posed by cyber attacks to cyber-physical systems (CPS) - concrete attacks at a technical level, e.g. on bus systems - security solutions specific to CPS with their capabilities and limitations - examples of security architectures for CPS and the requirements they guarantee - standard security engineering processes for CPS		
<i>Fertigkeiten</i>	The students are able to - identify security threats and assess the risks for a given CPS - apply attack toolkits to analyse a networked control system, and detect attacks beyond those taught in class - identify and apply security solutions suitable to the requirements - follow security engineering processes to develop a security architecture for a given CPS - recognize challenges and limitations, e.g. posed by novel types of attack		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	The students are able to - expertly discuss security risks and incidents of CPS and their mitigation in a solution-oriented fashion with experts and non-experts - foster a security culture with respect to CPS and the corresponding critical infrastructures		
<i>Selbstständigkeit</i>	The students are able to - follow up and critically assess current developments in the security of CPS including relevant security incidents - master a new topic within the area by self-study and self-initiated interaction with experts and peers.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : Wahlpflicht		

Lehrveranstaltung L2691: Security of Cyber-Physical Systems	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Fröschle
Sprachen	EN
Zeitraum	WiSe
Inhalt	Embedded systems in energy, production, and transportation are currently undergoing a technological transition to highly networked automated cyber-physical systems (CPS). Such systems are potentially vulnerable to cyber attacks, and these can have physical impact. In this course we investigate security threats, solutions and architectures that are specific to CPS. The topics are as follows: Fundamentals and motivating examples Networked and embedded control systems Bus system level attacks Intruder detection systems (IDS), in particular physics-based IDS System security architectures, including cryptographic solutions Adversarial machine learning attacks in the physical world Aspects of Location and Localization Wireless networks and infrastructures for critical applications Communication security architectures and remaining threats Intruder detection systems (IDS), in particular data-centric IDS Resilience against multi-instance attacks Security Engineering of CPS: Process and Norms
Literatur	Recent scientific papers and reports in the public domain.

Lehrveranstaltung L2692: Security of Cyber-Physical Systems	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Fröschle
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0926: Verteilte Algorithmen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Verteilte Algorithmen (L1071)	Vorlesung	2	3
Verteilte Algorithmen (L1072)	Hörsaalübung	2	3
Modulverantwortlicher	Prof. Volker Turau		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Algorithmen und Datenstrukturen - Verteilte Systeme - Diskrete Mathematik - Graphentheorie		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Studierende können die wichtigsten Abstraktion von Verteilten Algorithmen erklären (synchrones/asynchrones Model, nachrichtenbasierte und speicherbasierte Kommunikation, Randomisierung). Sie sind in der Lage, komplexitätsmaße für verteilte Algorithmen zu beschreiben (Runden-, Nachrichten- und Speicherkomplexität). Sie können Basisalgorithmen für die wichtigsten verteilten Probleme: Leader election, wechselseitiger Ausschluss, Graphfärbungen, Spannbäume beschreiben. Sie kennen die wesentlichen Techniken von radomisierten Algorithmen. <i>Fertigkeiten</i> Studierende können eigene verteilte Algorithmen entwerfen und der Komplexität analysieren. Sie greifen dabei auf existierende Standardalgorithmen zurück. Sie analysieren die Komplexität randomisierter Algorithmen. Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	45 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht		

Lehrveranstaltung L1071: Verteilte Algorithmen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Volker Turau
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	- Leader Election - Färbungen & Unabhängige Mengen - Algorithmen für Bäume - Minimal aufspannende Bäume - Randomisierte Verteilte Algorithmen - Wechselseitiger Ausschluss
Literatur	- David Peleg: Distributed Computing - A Locality-Sensitive Approach. SIAM Monograph, 2000 - Gerard Tel: Introduction to Distributed Algorithms, Cambridge University Press, 2nd edition, 2000 - Nancy Lynch: Distributed Algorithms. Morgan Kaufmann, 1996 - Volker Turau: Algorithmische Graphentheorie. Oldenbourg Wissenschaftsverlag, 3. Auflage, 2004.

Lehrveranstaltung L1072: Verteilte Algorithmen	
Typ	Hörsaalübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Volker Turau
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1400: Entwurf von Dependable Systems				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Entwurf von Dependable Systems (L2000)		Vorlesung	2	3
Entwurf von Dependable Systems (L2001)		Gruppenübung	2	3
Modulverantwortlicher		Prof. Görschwin Fey		
Zulassungsvoraussetzungen		Keine		
Empfohlene Vorkenntnisse		Grundlegende Kenntnisse zu Datenstrukturen und Algorithmen		
Modulziele/ angestrebte Lernergebnisse		Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz		<i>Wissen</i> Im Folgenden wird "Dependable" als Zusammenfassung von Zuverlässigkeit, Verfügbarkeit, Wartbarkeit, Sicherheit (Safety & Security) verwendet. Kenntnis von Ansätzen zum Entwurf von Dependable Systems, z.B. • Strukturelle Lösungen wie z.B. Modular Redundancy • Algorithmische Lösungen wie z.B. Behandlung Byzantinischer Fehler, Checkpointing, etc. Kenntnis von Methoden zur Analyse der Dependability von Systemen <i>Fertigkeiten</i> Fähigkeit zum Entwurf von Dependable Systems durch Implementierung der obigen Ansätze. Fähigkeit zur Analyse der Dependability von Systemen durch Anwendung der obigen Analysemethoden. Personale Kompetenzen <i>Sozialkompetenz</i> Studierende können • die jeweiligen Konzepte diskutieren und erläutern sowie • die Lösungen mündlich darstellen. <i>Selbstständigkeit</i> Studierende erlernen mittels Zusatzmaterial selbständig vertiefende Zusammenhänge der Konzepte aus der Vorlesung und erweiterte Lösungsverfahren.		
Arbeitsaufwand in Stunden		Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte		6		
Studienleistung		Verpflichtend Bonus	Art der Studienleistung	Beschreibung
		Ja Keiner	Fachtheoretisch- fachpraktische Studienleistung	Die Lösung einer Aufgabe ist Zulassungsvoraussetzung für die Prüfung. Die Aufgabe wird in Vorlesung und Übung definiert.
Prüfung		Mündliche Prüfung		
Prüfungsdauer und -umfang		30 min		
Zuordnung zu folgenden Curricula		Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme: Wahlpflicht Mechatronics: Vertiefung Systementwurf: Wahlpflicht Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht		

Lehrveranstaltung L2000: Entwurf von Dependable Systems	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Görschwin Fey
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Beschreibung Der Begriff „Dependability“ umfasst verschiedene Aspekte eines Systems. Dies sind typischer Weise: • Zuverlässigkeit • Verfügbarkeit • Wartbarkeit • Sicherheit - Safety & Security Damit ist Dependability ein zentraler Aspekt, der früh im Systementwurf betrachtet werden muss. Dies gilt für Software, Eingebettete Systeme wie auch umfassende Cyber-Physical Systems. Inhalt Das Modul führt grundlegende Konzept zum Entwurf und zur Analyse von Dependable Systems ein. Entwurfsbeispiele dienen dazu, eigene praktische Erfahrung zu sammeln. Ein Schwerpunkt des Moduls liegt im Bereich eingebetteter Systeme. Folgende Gebiete werden betrachtet: • Modellierung • Fehlertoleranz • Entwurfskonzepte • Analyse von Systemen
Literatur	

Lehrveranstaltung L2001: Entwurf von Dependable Systems	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Görschwin Fey
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1685: Ausgewählte Aspekte der Informatik**Lehrveranstaltungen**

Titel	Typ	SWS	LP
Ausgewählte Aspekte der Informatik (L2672)	Vorlesung	3	4
Ausgewählte Aspekte der Informatik (L2673)	Gruppenübung	1	2
Modulverantwortlicher	Prof. Görschwin Fey		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht		

Lehrveranstaltung L2672: Ausgewählte Aspekte der Informatik

Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	Dieses Modul vermittelt einen Einblick in ein aktuelles Thema der Informatik.
Literatur	

Lehrveranstaltung L2673: Ausgewählte Aspekte der Informatik

Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1682: Secure Software Engineering**Lehrveranstaltungen**

Titel	Typ	SWS	LP
Entwicklung von sicherer Software (L2667)	Vorlesung	2	3
Entwicklung von sicherer Software (L2668)	Projekt-/problembasierte Lehrveranstaltung	2	3
Modulverantwortlicher	Prof. Riccardo Scandariato		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht		

Lehrveranstaltung L2667: Secure Software Engineering

Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Riccardo Scandariato
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2668: Secure Software Engineering

Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Riccardo Scandariato
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Modul M1812: Constraint Satisfaction Problems			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Constraint Satisfaction Problems (L3002)	Vorlesung	2	3
Constraint Satisfaction Problems (L3003)	Hörsaalübung	2	3
Modulverantwortlicher	Prof. Antoine Mottet		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht		

Lehrveranstaltung L3002: Constraint Satisfaction Problems	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Antoine Mottet
Sprachen	EN
Zeitraum	SoSe
Inhalt	This course gives an introduction to the topic of constraint satisfaction problems and their complexity. It will cover the basics of the theory such as the universal-algebraic approach to constraint satisfaction and several classical algorithms such as local consistency checking and the Bulatov-Dalmau algorithm. We will finally discuss the recent research directions in the field. Educational Objectives: After taking part successfully, students have reached the following learning results Professional Competence:- Knowledge:* Students can describe basic concepts from the theory of constraint satisfaction such as primitive positive formulas, interpretations, polymorphisms, clones* Students can discuss the connections between these concepts* Students know proofs strategies and can reproduce them- Skills:* Students can use CSPs to model problems from complexity theory and decide their complexity using methods from the course.
Literatur	

Lehrveranstaltung L3003: Constraint Satisfaction Problems	
Typ	Hörsaalübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Antoine Mottet
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1886: GPU Arcuitectures and Programming			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
GPU Architectures and Programming (L3120)	Vorlesung	2	3
GPU Architectures and Programming (L3121)	Projekt-/problembasierte Lehrveranstaltung	2	3
Modulverantwortlicher	Prof. Sohan Lal		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : Wahlpflicht Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht		

Lehrveranstaltung L3120: GPU Architectures and Programming	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sohan Lal
Sprachen	EN
Zeitraum	SoSe
Inhalt	- Review of computer architecture basics - measuring performance, benchmarks, five-stage RISC pipeline, caches - GPU basics - evolution of GPU computing, a high-level overview of a GPU architecture - GPU programming with CUDA - program structure, CUDA threads organization, warp/thread-block scheduling - GPU (micro) architecture - streaming multiprocessors, single instruction multiple threads (SIMT) core design, tensor/RT cores, mixed-precision support - GPU memory hierarchy - banked register file and operand collectors, shared memory, GPU caches (differences w.r.t. CPU caches), global memory - Branch and memory divergence - branch handling, stack-based reconvergence, memory coalescing, coalescer design - Barriers and synchronization - Temporal and spatial locality exploitation challenges in GPU caches - Global memory- high throughput requirements, GDDR/HBM, memory bandwidth optimization techniques - GPU research issues - performance bottlenecks, GPU power modeling, high-power consumption/energy efficiency, GPU security - Application case study - deep learning - Cycle-accurate simulators for GPUs The learning in the lectures will be augmented by a semester-long problem-based project.
Literatur	• David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors - A Hands-on Approach, Second Edition (Book) • David A. Patterson and John L. Hennessy, Computer Architecture: A Quantitative Approach, 5th Edition (Book)

Lehrveranstaltung L3121: GPU Architectures and Programming	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sohan Lal
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1397: Modellprüfung - Beweiser und Algorithmen				
Lehrveranstaltungen				
Titel	Typ		SWS	LP
Modellprüfung - Beweiser und Algorithmen (L1979)	Vorlesung		2	3
Modellprüfung - Beweiser und Algorithmen (L1980)	Gruppenübung		2	3
Modulverantwortlicher	Prof. Görschwin Fey			
Zulassungsvoraussetzungen	Keine			
Empfohlene Vorkenntnisse	Grundlegende Kenntnisse zu Datenstrukturen und Algorithmen			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz <i>Wissen</i>	Studierende kennen - Algorithmen und Datenstrukturen für die Modellprüfung, - grundlegende Beweisverfahren sowie - den Einfluss der Modellierung und Spezifikation auf den Rechenaufwand für den Nachweis mittels Modellprüfung.			
<i>Fertigkeiten</i>	Studierende können - Algorithmen und Datenstrukturen zur Modellprüfung erläutern und implementieren - abschätzen, ob sich eine Problemstellung mittels Boolescher Beweisverfahren oder Modellprüfung beantworten lässt, und - solche Lösungsverfahren realisieren.			
Personale Kompetenzen <i>Sozialkompetenz</i>	Studierende können - die jeweiligen Konzepte diskutieren und erläutern sowie - die Lösungen mündlich darstellen.			
<i>Selbstständigkeit</i>	Studierende erlernen mittels Zusatzmaterial selbständig vertiefende Zusammenhänge der Konzepte aus der Vorlesung und erweiterte Lösungsverfahren.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Verpflichtend Ja	Bonus Keiner	Art der Studienleistung Fachtheoretisch-fachpraktische Studienleistung	Beschreibung Die Aufgabe wird im Rahmen von Vorlesung und Prüfung definiert. Die Lösung der Aufgabe ist Zulassungsvoraussetzung für die Prüfung.
Prüfung	Mündliche Prüfung			
Prüfungsdauer und -umfang	30 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme: Wahlpflicht			

Lehrveranstaltung L1979: Modellprüfung - Beweiser und Algorithmen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Görschwin Fey
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Correctness is a major concern in embedded systems. Model checking can fully automatically proof formal properties about digital hardware or software. Such properties are given in temporal logic, e.g., to prove "No two orthogonal traffic lights will ever be green." And how do the underlying reasoning algorithms work so effectively in practice despite a computational complexity of NP hardness and beyond? But what are the limitations of model checking? How are the models generated from a given design? The lecture will answer these questions. Open source tools will be used to gather a practical experience. Among other topics, the lecture will consider the following topics: • Modelling digital Hardware, Software, and Cyber Physical Systems • Data structures, decision procedures and proof engines ◦ Binary Decision Diagrams ◦ And-Inverter-Graphs ◦ Boolean Satisfiability ◦ Satisfiability Modulo Theories • Specification Languages ◦ CTL ◦ LTL ◦ System Verilog Assertions • Algorithms for ◦ Reachability Analysis ◦ Symbolic CTL Checking ◦ Bounded LTL-Model Checking ◦ Optimizations, e.g., induction, abstraction • Quality assurance
Literatur	Edmund M. Clarke, Jr., Orna Grumberg, and Doron A. Peled. 1999. <i>Model Checking</i>. MIT Press, Cambridge, MA, USA. A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh. 2009. <i>Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications</i>. IOS Press, Amsterdam, The Netherlands, The Netherlands. Selected research papers

Lehrveranstaltung L1980: Modellprüfung - Beweiser und Algorithmen	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Görschwin Fey
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1301: Software Testing				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Softwaretesten (L1791)		Vorlesung	2	3
Softwaretesten (L1792)		Projekt-/problembasierte Lehrveranstaltung	2	3
Modulverantwortlicher		Prof. Sibylle Schupp		
Zulassungsvoraussetzungen		None		
Empfohlene Vorkenntnisse		• Software Engineering • Higher Programming Languages • Object-Oriented Programming • Algorithms and Data Structures • Experience with (Small) Software Projects • Statistics		
Modulziele/ angestrebte Lernergebnisse		Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz		Students explain the different phases of testing, describe fundamental techniques of different types of testing, and paraphrase the basic principles of the corresponding test process. They give examples of software development scenarios and the corresponding test type and technique. They explain algorithms used for particular testing techniques and describe possible advantages and limitations.		
<i>Wissen</i>				
Fertigkeiten		Students identify the appropriate testing type and technique for a given problem. They adapt and execute respective algorithms to execute a concrete test technique properly. They interpret testing results and execute corresponding steps for proper re-test scenarios. They write and analyze test specifications. They apply bug finding techniques for non-trivial problems.		
<i>Fertigkeiten</i>				
Personale Kompetenzen		Students discuss relevant topics in class. They defend their solutions orally. They communicate in English.		
<i>Sozialkompetenz</i>				
<i>Selbstständigkeit</i>		Students can assess their level of knowledge continuously and adjust it appropriately, based on feedback and on self-guided studies. Within li own learning goals. Upon successful completion, students can identify and precisely formulate new problems in academic or applied research testing. Within this field, they can conduct independent studies to acquire the necessary competencies and compile their findings in acad devise plans to arrive at new solutions or assess existing ones		
Arbeitsaufwand in Stunden		Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte		6		
Studienleistung		Keine		
Prüfung		Fachtheoretisch-fachpraktische Arbeit		
Prüfungsdauer und -umfang		Software		
Zuordnung zu folgenden Curricula		Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : W		

Lehrveranstaltung L1791: Software Testing	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	• Fundamentals of software testing • Model-based testing • Test automation • Criteria-based testing
Literatur	• M. Pezze and M. Young, Software Testing and Analysis, John Wiley 2008. • P. Ammann and J. Offutt, "Introduction to Software Testing", 2nd edition 2016. • A. Zeller: "Why Programs Fail: A Guide to Systematic Debugging", 2nd edition 2012.

Lehrveranstaltung L1792: Software Testing	
Typ	Projekt-/problem-basierte Lehrveranstaltung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	• Fundamentals of software testing • Model-based testing • Test automation • Criteria-based testing
Literatur	• M. Pezze and M. Young, Software Testing and Analysis, John Wiley 2008. • P. Ammann and J. Offutt, "Introduction to Software Testing", 2nd edition 2015.

Modul M1427: Algorithmische Spieltheorie			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Algorithmische Spieltheorie (L2060)	Vorlesung	2	4
Algorithmische Spieltheorie (L2061)	Hörsaalübung	2	2
Modulverantwortlicher	Prof. Matthias Mnich		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Mathematik I - Mathematik II - Algorithmen und Datenstrukturen		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>	- Die Studierenden können die Grundkonzepte der algorithmischen Spieltheorie und des Mechanismusdesigns benennen. Sie können sie anhand geeigneter Beispiele erklären. - Die Studierenden können logische Zusammenhänge zwischen diesen Konzepten diskutieren. Sie können diese Zusammenhänge anhand von Beispielen veranschaulichen. - Sie kennen Spiel- und Mechanismusdesignstrategien und können diese reproduzieren. - Die Studierenden können mithilfe der in diesem Kurs untersuchten Konzepte strategische Interaktionssysteme von Agenten modellieren. Darüber hinaus sind sie in der Lage, ihre Effizienz und Gleichgewichte mithilfe etablierter Methoden zu analysieren. - Die Studierenden können weitere logische Zusammenhänge zwischen den im Kurs untersuchten Konzepten entdecken und überprüfen. - Für ein bestimmtes Problem können die Studierenden einen geeigneten Ansatz entwickeln und ausführen und die Ergebnisse kritisch bewerten. - Die Studierenden können in Teams zusammenarbeiten. Sie können Mathematik als gemeinsame Sprache verwenden. - Auf diese Weise können sie neue Konzepte entsprechend den Bedürfnissen ihrer Kooperationspartner kommunizieren. Darüber hinaus können sie Beispiele entwerfen, um das Verständnis ihrer Kommilitonen zu überprüfen und zu vertiefen. - Die Studierenden können ihr Verständnis komplexer Konzepte selbst überprüfen. Sie können offene Fragen genau spezifizieren und wissen, wo sie Hilfe bei der Lösung erhalten können. - Die Studierenden haben eine ausreichende Ausdauer entwickelt, um länger zielgerichtet an schwierigen Problemen arbeiten zu können.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht		

Lehrveranstaltung L2060: Algorithmische Spieltheorie	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Prof. Matthias Mnich
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Algorithmische Spieltheorie ist ein Thema an der Schnittstelle von Informatik, Wirtschaft und Mathematik. Es befasst sich mit der Analyse des Verhaltens und der Interaktionen strategischer Akteure, die häufig versuchen, ihre Anreize zu maximieren. Die Umgebung, in der diese Agenten interagieren, wird als Spiel bezeichnet. Wir möchten verstehen, ob die Agenten ein "Gleichgewicht" oder einen stabilen Zustand des Spiels erreichen können, in dem die Agenten keinen Anreiz haben, von ihren gewählten Strategien abzuweichen. Der algorithmische Teil besteht darin, effiziente Methoden zu entwickeln, um Gleichgewichte in Spielen zu finden, und den Agenten Empfehlungen zu geben, damit sie schnell einen Zustand persönlicher Zufriedenheit erreichen können. Wir werden auch das Mechanismusdesign untersuchen. Beim Mechanismusdesign möchten wir Märkte und Auktionen gestalten und den Agenten strategische Optionen geben, damit sie einen Anreiz haben, rational zu handeln. Wir möchten die Märkte und Auktionen auch so gestalten, dass sie effizient sind, dass alle Waren freigegeben werden und die Agenten die von ihnen erworbenen Waren nicht überbezahlen. Themen: • grundlegende Gleichgewichtskonzepte (Nash-Gleichgewichte, korrelierte Gleichgewichte, ...) • strategische Maßnahmen (Best-Response-Dynamik, No-Regret-Dynamik, ...) • Auktionsdesign (umsatzmaximierende Auktionen, Vickrey-Auktionen) • stabile Matching-Theorie (Präferenzaggregationen, Nierenaustausch, ...) • Preis der Anarchie und egoistisches Routing (Braess 'Paradoxon, Überlastungsspiele, ...)
Literatur	• T. Roughgarden: Twenty Lectures on Algorithmic Game Theory, Cambridge University Press, 2016. • N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani. Algorithmic Game Theory. Cambridge University Press, 2007.

Lehrveranstaltung L2061: Algorithmische Spieltheorie	
Typ	Hörsaalübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Matthias Mnich
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1248: Compiler für Eingebettete Systeme			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Compiler für Eingebettete Systeme (L1692)	Vorlesung	3	4
Compiler für Eingebettete Systeme (L1693)	Projekt-/problembasierte Lehrveranstaltung	1	2
Modulverantwortlicher	Prof. Heiko Falk		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Modul "Eingebettete Systeme" C/C++ Programmierkenntnisse		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i>	Die Bedeutung Eingebetteter Systeme steigt von Jahr zu Jahr. Innerhalb Eingebetteter Systeme steigt der Software-Anteil, der auf Prozessoren ausgeführt wird, aufgrund geringerer Kosten und höherer Flexibilität ebenso kontinuierlich. Wegen der besonderen Einsatzgebiete Eingebetteter Systeme kommen hier hochgradig spezialisierte Prozessoren zum Einsatz, die applikationsspezifisch auf ihr jeweiliges Einsatzgebiet ausgerichtet sind. Diese hochgradig spezialisierten Prozessoren stellen hohe Anforderungen an einen Compiler, der Code von hoher Qualität generieren soll. Nach erfolgreichem Besuch der Veranstaltung sind die Studierenden in der Lage, • Struktur und Aufbau derartiger Compiler aufzuzeigen, • interne Zwischendarstellungen auf verschiedenen Abstraktionsniveaus zu unterscheiden und zu erklären, und • Probleme und Optimierungen in allen Compilerphasen zu beurteilen. Wegen der hohen Anforderungen an Compiler für Eingebettete Systeme sind effektive Optimierungen unerlässlich. Die Studierenden lernen insbes., • welche Arten von Optimierungen es auf Quellcode-Niveau gibt, • wie die Übersetzung von der Quellsprache nach Assembler abläuft, • welche Arten von Optimierungen auf Assembler-Niveau durchzuführen sind, • wie die Registerallokation vonstatten geht, und • wie Speicherhierarchien effizient ausgenutzt werden. Da Compiler für Eingebettete Systeme oft verschiedene Zielfunktionen optimieren sollen (z.B. durchschnittliche oder worst-case Laufzeit, Energieverbrauch, Code-Größe), lernen die Studierenden den Einfluss von Optimierungen auf diese verschiedenen Zielfunktionen zu beurteilen.		
<i>Fertigkeiten</i>	Studierende werden in die Lage versetzt, hochsprachlichen Programmcode in Maschinensprache zu übersetzen. Die Studierenden erwerben die Fähigkeit zu beurteilen, welche Art von Code-Optimierung innerhalb eines Compilers am effektivsten auf welchem Abstraktionsniveau (bspw. Quell- oder Assemblercode) durchzuführen ist. Während der Übungen erwerben die Studierenden die Fähigkeit, einen funktionierenden Compiler mitsamt Optimierungen zu implementieren.		
Personale Kompetenzen <i>Sozialkompetenz</i>	Die Studierenden sind nach Abschluss des Moduls in der Lage, ähnliche Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.		
<i>Selbstständigkeit</i>	Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Nachrichten- und Kommunikationstechnik: Wahlpflicht Flugzeug-Systemtechnik: Kernqualifikation: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mechatronics: Vertiefung Systementwurf: Wahlpflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht		

Lehrveranstaltung L1692: Compiler für Eingebettete Systeme	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	• Einleitung und Motivation • Compiler für Eingebettete Systeme - Anforderungen und Abhängigkeiten • Interne Struktur von Compilern • Pre-Pass Optimierungen • HIR Optimierungen und Transformationen • Code-Generierung • LIR Optimierungen und Transformationen • Register-Allokation • WCET-bewusste Code-Generierung • Ausblick
Literatur	• Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2nd Edition, Springer, 2012. • Steven S. Muchnick. Advanced Compiler Design and Implementation. Morgan Kaufmann, 1997. • Andrew W. Appel. Modern compiler implementation in C. Oxford University Press, 1998.

Lehrveranstaltung L1693: Compiler für Eingebettete Systeme	
Typ	Projekt-/problemorientierte Lehrveranstaltung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0556: Computer Graphics				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Computer-Grafik (L0145)		Vorlesung	2	3
Computer-Grafik (L0768)		Gruppenübung	2	3
Modulverantwortlicher	Prof. Tobias Knopp			
Zulassungsvoraussetzungen	None			
Empfohlene Vorkenntnisse	• Linear Algebra (in particular matrix/vector computation) • Basic programming skills in C/C++			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz	<i>Wissen</i> Students can explain and describe basic algorithms in 3D computer graphics. <i>Fertigkeiten</i> Students are capable of • implementing a basic 3D rendering pipeline. This consists of projecting simple 3D structures (e.g. cube, spheres) onto a 2D surface using a virtual camera. • apply geometric transformations (e.g. rotation, scaling) in 2D and 3D computer graphics. • using well-known 2D/3D APIs (OpenGL, Cairo) for solving a given problem statement. Personale Kompetenzen <i>Sozialkompetenz</i> Students can collaborate in a small team on the realization and validation of a 3D computer graphics pipeline. <i>Selbstständigkeit</i> • Students are able to solve simple tasks independently with reference to the contents of the lectures and the exercise sets. • Students are able to solve detailed problems independently with the aid of the tutorial's programming task.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Keine			
Prüfung	Klausur			
Prüfungsdauer und -umfang	90 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Signalverarbeitung: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Informationstechnologie: Wahlpflicht			

Lehrveranstaltung L0145: Computer Graphics	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	EN
Zeitraum	SoSe
Inhalt	Computer graphics and animation are leading to an unprecedented visual revolution. The course deals with its technological foundations: • Object-oriented Computer Graphics • Projections and Transformations • Polygonal and Parametric Modelling • Illuminating, Shading, Rendering • Computer Animation Techniques • Kinematics and Dynamics Effects Students will be working on a series of mini-projects which will eventually evolve into a final project. Learning computer graphics and animation resembles learning a musical instrument. Therefore, doing your projects well and in time is essential for performing well on this course.
Literatur	Alan H. Watt: 3D Computer Graphics. Harlow: Pearson (3rd ed., repr., 2009). Dariush Derakhshani: Introducing Autodesk Maya 2014. New York, NY : Wiley (2013).

Lehrveranstaltung L0768: Computer Graphics	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1741: Betriebssystembau			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Betriebssystembau (L2812)	Vorlesung	2	3
Betriebssystembau (L2814)	Projekt-/problembasierte Lehrveranstaltung	2	2
Betriebssystembau (L2813)	Hörsaalübung	1	1
Modulverantwortlicher	Prof. Christian Dietrich		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	• Objekt-orientiertes Programmieren (notwendig) • Grundlagen von Betriebssystemen (empfohlen) • Grundlagen der Rechnerarchitektur (empfohlen) • Programmieren in C/C++ (empfohlen)		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • erläutern den Startvorgang eines Rechensystems am Beispiel eines IA32 PCs. • beschreiben die spezifischen Herausforderungen bei der Softwareentwicklung für "bare metal". • beschreiben den Ablauf einer Unterbrechungsbehandlung von der Hardware bis zur (System-)software. • skizzieren Besonderheiten und Strategien der Unterbrechungsbehandlung in Hardware für Mehrkernsystemen am Beispiel des IA32-APICs. • unterscheiden die verschiedenen Typen von Kontrollflüssen in einem Betriebssystem anhand des Ebenenmodells. • unterscheiden harte, mehrstufige, und weiche Verfahren zur Unterbrechungssynchronisation in Betriebssystemen. • analysieren das Zusammenspiel von Scheduling und Unterbrechungssynchronisation. • unterscheiden grundlegende Möglichkeiten der Koordinierung und Synchronisation von Fäden (aktives/passives Warten, nichtverdrängbare kritische Abschnitte). • kennen die grundlegenden Synchronisationsprobleme (lost update, lost wakeup) und schlagen geeignete Gegenmaßnahmen vor. • können verschiedene Treibermodelle unterscheiden. • vergleichen grundlegende BS-Architekturen (Bibliothek, Monolith, Mikrokern, Exokern, Hypervisor) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und Mechanismen. • schildern die grundlegenden Paradigmen zur Interprozesskommunikation in Betriebssystemen (speicherbasiert vs. nachrichtenbasiert).		
<i>Fertigkeiten</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • diskutieren die Aufgabenteilung zwischen Hardware und Systemsoftware bei der Unterbrechungsbearbeitung. • können mehrstufige Verfahren zur Unterbrechungssynchronisation implementieren. • klassifizieren konkrete Konkurrenzsituationen und leiten daraus geeignete Synchronisationsmaßnahmen ab. • entwickeln den Koroutinenwechsel für einen gegebene Architektur. • können verdrängendes Scheduling in einem Betriebssystem implementieren. • entwickeln Mechanismen für die Synchronisation auf Fadenebene. • können Gerätetreiber in eine Betriebssystemarchitektur einbinden. • skizzieren, wie höhere Synchronisationskonstrukte aus grundlegenden Synchronisationsprimitiven implementiert werden (Monitore, Leser-/Schreiber-Sperre). • können Primitiven zur Interprozesskommunikation implementieren und einsetzen.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • können in Kleingruppen kooperativ arbeiten. • können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.		
<i>Selbstständigkeit</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • können sich komplexe Fehlerbilder durch methodisches Vorgehen schrittweise erschließen. • reflektieren ihre Entscheidungen kritisch und leiten Alternativen ab. • können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen. • können getroffene Fehlentscheidungen revidieren bzw. die entstehenden Kosten bewusst in Kauf nehmen.		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus Nein 20 %	Art der Studienleistung	Beschreibung
		Fachtheoretisch-fachpraktische Studienleistung	
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	25 min		

Zuordnung zu folgenden	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht
Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht

Lehrveranstaltung L2812: Betriebssystembau	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Christian Dietrich
Sprachen	DE
Zeitraum	SoSe
Inhalt	Ziel der Vorlesung ist die Vermittlung von konzeptionellen Grundlagen und wichtigen Techniken, die für den Bau eines Betriebssystems erforderlich sind. Dabei werden gleichzeitig Grundlagen aus dem Betriebssystembereich wie Unterbrechungen, Synchronisation und Ablaufplanung, die aus anderen Veranstaltungen weitgehend bekannt sein sollten, wiederholt und vertieft. • Grundlagen der Betriebssystementwicklung • Unterbrechungen (Hardware, Software, Synchronisation) • IA-32: Die 32-Bit-Intel-Architektur • Koroutinen und Programmfäden • Scheduling • Betriebssystem-Architekturen • Fadensynchronisation • Gerätetreiber • Interprozesskommunikation
Literatur	

Lehrveranstaltung L2814: Betriebssystembau	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Christian Dietrich
Sprachen	DE
Zeitraum	SoSe
Inhalt	Ziel ist es, schrittweise ein kleines Betriebssystem für den PC zu entwickeln. Für die Bearbeitung der Übungsaufgaben werden Gruppen gebildet. Dabei gibt es zwei Varianten der Übung: • OOSTuBS ist die klassische Uniprozessorvariante • MPSTuBS ist die Variante für moderne Mehrkernrechner Die Veranstaltung ist in mehrere Aufgaben gegliedert, die von den Gruppen bearbeitet werden. Dabei arbeiten die Gruppen die gesamte Zeit an "ihrem" Betriebssystem, und erhalten durch die Betreuer für jede neue Aufgabe nur eine ergänzende Vorgabe.
Literatur	

Lehrveranstaltung L2813: Betriebssystembau	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Christian Dietrich
Sprachen	DE
Zeitraum	SoSe
Inhalt	In den Hörsaalübungen werden die abstrakt vorgestellten Konzepte aus der Vorlesung anhand einer konkreten Rechnerarchitektur (z.B. IA-32) vertieft. Dies findet dabei mit Hinblick auf die praktischen Aufgaben der begleitenden problembasierten Lehrveranstaltung statt und stützt die Studierenden mit den dafür nötigen Kenntnissen aus. • Grundlegende Ein-/Ausgabekanäle • Behandlung von Hardwareunterbrechungen • Harte Synchronisationsprimitiven • Pro-/Epilogmodell zur Interruptsynchronisation • Kontextwechsel und Kooperatives Scheduling • Programmierung des Zeitgebers • Übergang von und zum Leerlaufmodus des Betriebssystems • Implementierung von Semaphoren • Beenden von Fäden in einem Mehrkernsystem
Literatur	

Modul M1810: Autonomous Cyber-Physical Systems				
Lehrveranstaltungen				
Titel	Typ		SWS	LP
Autonomous Cyber-Physical Systems (L3000)	Vorlesung		2	3
Autonomous Cyber-Physical Systems (L3001)	Gruppenübung		2	3
Modulverantwortlicher	Prof. Bernd-Christian Renner			
Zulassungsvoraussetzungen	None			
Empfohlene Vorkenntnisse	• Very Good knowledge and practical experience in programming in the C language (Module: Procedural Programming) • Basic knowledge in software engineering • Basic knowledge in wired and wireless communication protocols • Principal understanding of simple electronic circuits			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz				
Wissen				
Fertigkeiten				
Personale Kompetenzen				
Sozialkompetenz				
Selbstständigkeit				
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung	
	Nein 10 %	Testate		
Prüfung	Klausur			
Prüfungsdauer und -umfang	90 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : Wahlpflicht			

Lehrveranstaltung L3000: Autonomous Cyber-Physical Systems	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Bernd-Christian Renner
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L3001: Autonomous Cyber-Physical Systems	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Bernd-Christian Renner
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1774: Advanced Internet Computing			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Advanced Internet Computing (L2916)	Vorlesung	2	3
Advanced Internet Computing (L2917)	Projekt-/problembasierte Lehrveranstaltung	2	3
Modulverantwortlicher	Prof. Stefan Schulte		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Good programming skills are necessary. Previous knowledge in the field of distributed systems is helpful.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	After successful completion of the course, students are able to:		
<i>Wissen</i>	- Describe basic concepts of Cloud Computing, the Internet of Things (IoT), and blockchain technologies - Discuss and assess critical aspects of Cloud Computing, the IoT, and blockchain technologies - Select and apply cloud and IoT technologies for particular application areas - Design and develop practical solutions for the integration of smart objects in IoT, Cloud, and blockchain software - Implement IoT services		
<i>Fertigkeiten</i>	The students acquire the ability to model Internet-based distributed systems and to work with these systems. This comprises especially the ability to select and utilize fitting technologies for different application areas. Furthermore, students are able to critically assess the chosen technologies.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Students can work on complex problems both independently and in teams. They can exchange ideas with each other and use their individual strengths to solve the problem.		
<i>Selbstständigkeit</i>	Students are able to independently investigate a complex problem and assess which competencies are required to solve it.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Netze: Wahlpflicht		

Lehrveranstaltung L2916: Advanced Internet Computing	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Stefan Schulte
Sprachen	EN
Zeitraum	SoSe
Inhalt	This lecture discusses modern Internet-based distributed systems in three blocks: (i) Cloud computing, (ii) the Internet of Things, and (iii) blockchain technologies. The following topics will be covered in the single lectures: - Cloud Computing - Elastic Computing - Technologies for identification for the IoT: RFID & EPC - Communication in the IoT: Standards and protocols - Security and trust in the IoT: Concerns and solution approaches - Edge and Fog Computing - Application areas: Smart factories, smart cities, smart healthcare - Blockchain technologies - Consensus
Literatur	Will be discussed in the lecture

Lehrveranstaltung L2917: Advanced Internet Computing	
Typ	Projekt-/problemorientierte Lehrveranstaltung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Stefan Schulte
Sprachen	EN
Zeitraum	SoSe
Inhalt	This project-/problemoriented part of the module augments the theoretical content of the lecture by a concrete technical problem, which needs to be solved by the students in group work during the semester. Possible topics are (blockchain-based) sensor data integration, Big Data processing, Cloud-based redundant data storages, and Cloud-based Onion Routing.
Literatur	Will be discussed in the lecture.

Modul M0924: Software für Eingebettete Systeme			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Software für Eingebettete Systeme (L1069)	Vorlesung	2	3
Software für Eingebettete Systeme (L1070)	Gruppenübung	3	3
Modulverantwortlicher	Prof. Bernd-Christian Renner		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	• Sehr gute Kenntnisse und Erfahrung in Programmiersprache C • Grundkenntnisse in Softwaretechnik • Prinzipielles Verständnis von Assembler Sprachen		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Studierende können die grundlegende Prinzipien und Vorgehensweisen für die Erstellung von Software für eingebettete Systeme erklären. Sie sind in der Lage, ereignisbasierte Programmier Techniken mittels Interrupts zu beschreiben. Sie kennen den Aufbau und Funktion eines konkreten Mikrocontrollers. Die Teilnehmer sind in der Lage, Anforderungen an Echtzeitsysteme zu erläutern. Sie können mindestens drei Scheduling Algorithmen für Echtzeitbetriebssysteme erläutern (einschließlich Vor- und Nachteile) <i>Fertigkeiten</i> Studierende erstellen interrupt-basierte Programme für einen konkreten Mikrocontroller. Sie erstellen und benutzen einen preemptiven scheduler. Sie setzen periphere Komponenten (Timer, ADCs, EEPROM) für komplexe Aufgaben eingebetteter System ein. Für den Anschluss externer Komponenten setzen sie serielle Protokolle ein. Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Nein 10 %	Testate	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Nachrichten- und Kommunikationstechnik: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mechatronics: Vertiefung Systementwurf: Wahlpflicht Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht		

Lehrveranstaltung L1069: Software für Eingebettete Systeme	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Bernd-Christian Renner
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	• General-Purpose Processors • Programming the Atmel AVR • Interrupts • C für Embedded Systems • Standard Single Purpose Processors: Peripherals • Finite-State Machines • Speicher • Betriebssystem für Eingebettete Systeme • Echtzeit Eingebettete Systeme
Literatur	1. Embedded System Design, F. Vahid and T. Givargis, John Wiley 2. Programming Embedded Systems: With C and Gnu Development Tools, M. Barr and A. Massa, O'Reilly 3. C und C++ für Embedded Systems, F. Bollow, M. Homann, K. Köhn, MITP 4. The Art of Designing Embedded Systems, J. Ganssle, Newnes 5. Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie, G. Schmitt, Oldenbourg 6. Making Embedded Systems: Design Patterns for Great Software, E. White, O'Reilly

Lehrveranstaltung L1070: Software für Eingebettete Systeme	
Typ	Gruppenübung
SWS	3
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 48, Präsenzstudium 42
Dozenten	Prof. Bernd-Christian Renner
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0910: Fortgeschrittener Entwurf von Chip-Systemen (Praktikum)			
Lehrveranstaltungen			
Titel	Fortgeschrittener Entwurf von Chip-Systemen (L1061)	Typ	Projekt-/problembasierte Lehrveranstaltung
		SWS	3
		LP	6
Modulverantwortlicher	Prof. Heiko Falk		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Erfolgreiche Teilnahme am praktischen FPGA-Labor des Moduls "Rechnerarchitektur" ist zwingende Voraussetzung.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> In diesem Modul werden fortgeschrittene Konzepte der Rechnerarchitektur praxisorientiert vermittelt. Mit Hilfe der Hardware-Beschreibungssprache VHDL und rekonfigurierbarer FPGA-Hardware lernen Studierende, wie komplexe Rechensysteme (sog. Systems-on-Chip, SoCs), wie sie insbesondere im Bereich der eingebetteten Systeme anzutreffen sind, in Hardware zu entwerfen sind. Ausgehend von einer einfachen Prozessor-Architektur lernen Studierende, die Verarbeitung von Befehlen durch eine Maschine nach dem Pipelining-Prinzip zu realisieren. Sie implementieren verschiedene Formen Cache-basierter Speicher-Hierarchien, untersuchen Ansätze zum dynamischen Scheduling von Maschinenbefehlen und zur Sprungvorhersage, und konstruieren letztlich ein komplexes MPSoC-System (multi-processor system-on-chip), das aus mehreren Kernen besteht, die über einen gemeinsamen Bus verbunden sind. <i>Fertigkeiten</i> Die Studierenden können analysieren, wie hochspezifische und individuelle Rechner aus einer Sammlung gängiger Einzelkomponenten zusammengesetzt werden. Sie sind in der Lage, die Wechselwirkungen zwischen einem physischen Rechensystem und der darauf ausgeführten Software beurteilen zu können. Sie sollen so in die Lage versetzt werden, Auswirkungen hardwarenaher Entwurfsentscheidungen auf die Leistung des Gesamtsystems abzuschätzen, zu beurteilen und geeignete Optionen vorzuschlagen. Personale Kompetenzen <i>Sozialkompetenz</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, ähnliche Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren. <i>Selbstständigkeit</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbstständig zu erarbeiten, das erworbene Wissen in konkrete Implementierungen komplexer Hardware-Strukturen zu überführen und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.		
Arbeitsaufwand in Stunden	Eigenstudium 138, Präsenzstudium 42		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Fachtheoretisch-fachpraktische Arbeit		
Prüfungsdauer und -umfang	VHDL-Code und FPGA-basierte Implementierungen		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht		

Lehrveranstaltung L1061: Fortgeschrittener Entwurf von Chip-Systemen	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	3
LP	6
Arbeitsaufwand in Stunden	Eigenstudium 138, Präsenzstudium 42
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	• Einführung in grundlegende Technologien (FPGAs, MIPS Einzelzyklus-Maschine) • Fließband-Befehlsverarbeitung • Cache-basierte Speicher-Hierarchien • Busse und Bus-Arbitrierung • Multi-Prozessor Chip-Systeme • Optional: Fortgeschrittene Fließband-Konzepte (Dynamisches Scheduling, Sprungvorhersage)
Literatur	• D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005. • A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001. • A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000.

Modul M0839: Traffic Engineering			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Seminar Traffic Engineering (L0902)	Seminar	2	2
Traffic Engineering (L0900)	Vorlesung	2	2
Traffic Engineering Übung (L0901)	Gruppenübung	1	2
Modulverantwortlicher	Prof. Andreas Timm-Giel		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	- Fundamentals of communication or computer networks - Stochastics		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Students are able to describe methods for planning, optimisation and performance evaluation of communication networks.		
<i>Fertigkeiten</i>	Students are able to solve typical planning and optimisation tasks for communication networks. Furthermore they are able to evaluate the network performance using queuing theory. Students are able to apply independently what they have learned to other and new problems. They can present their results in front of experts and discuss them.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>			
<i>Selbstständigkeit</i>	Students are able to acquire the necessary expert knowledge to understand the functionality and performance of new communication networks independently.		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Nachrichten- und Kommunikationstechnik: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Netze: Wahlpflicht		

Lehrveranstaltung L0902: Seminar Traffic Engineering	
Typ	Seminar
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Andreas Timm-Giel, Dr. Phuong Nga Tran
Sprachen	EN
Zeitraum	WiSe
Inhalt	Selected applications of methods for planning, optimization, and performance evaluation of communication networks, which have been introduced in the traffic engineering lecture are prepared by the students and presented in a seminar.
Literatur	U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Vieweg + Teubner - further literature announced in the lecture

Lehrveranstaltung L0900: Traffic Engineering	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Andreas Timm-Giel, Dr. Phuong Nga Tran
Sprachen	EN
Zeitraum	WiSe
Inhalt	Network Planning and Optimization • Linear Programming (LP) • Network planning with LP solvers • Planning of communication networks Queueing Theory for Communication Networks • Stochastic processes • Queueing systems • Switches (circuit- and packet switching) • Network of queues
Literatur	Literatur: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben / Literature: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer further literature announced in the lecture

Lehrveranstaltung L0901: Traffic Engineering Exercises	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Andreas Timm-Giel
Sprachen	EN
Zeitraum	WiSe
Inhalt	Accompanying exercise for the traffic engineering course
Literatur	Literatur: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer Weitere Literatur wird in der Lehrveranstaltung bekanntgegeben / Literature: U. Killat, Entwurf und Analyse von Kommunikationsnetzen, Springer further literature announced in the lecture

Modul M1742: Betriebssystemtechnik			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Betriebssystemtechnik (L2815)	Vorlesung	1	2
Betriebssystemtechnik (L2816)	Projekt-/problembasierte Lehrveranstaltung	3	4
Modulverantwortlicher	Prof. Christian Dietrich		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	• Objekt-orientiertes Programmieren (notwendig) • Programmieren in C/C++ (notwendig) • Betriebssystembau (empfohlen) • Grundlagen der Rechnerarchitektur (empfohlen)		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile. • klassifizieren Schutz-, Verwaltungs-, und Virtualisierungstechniken für Programmmzustände (Seitennummerierung, Segmentierung, Sprachbasierung, Capabilities) und implementieren diese auf der IA-32-Architektur • vergleichen grundlegende BS-Architekturen (Monolith, Mikrokern, Makrokern, Exokern) anhand fundamentaler Charakteristika (Robustheit, Performanz, Portierbarkeit) und deren Einfluss auf die Implementierung von Mechanismen (Systemaufrufe, Adressraumschutz). • diskutieren Adressraummodelle (Mehradressraummodell, Einadressraummodell, mehrstufige und inverse Seitenabbildungen, Mitbenutzung) und deren Implementierbarkeit auf gängigen Hardwarearchitekturen. • diskutieren Prinzipien der Mitbenutzung von Code und Daten unter Berücksichtigung der Betriebssystem- und Adressraumarchitektur. • können logischen, virtuellen und physischen Speicher voneinander abgrenzen. • können die Kostenvorteile von Zero-Copy Schnittstellen herleiten • können technische und konzeptionelle Sichtweise bei der Prozesserzeugung durch fork() voneinander abgrenzen.		
<i>Fertigkeiten</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • erläutern und implementieren Entwurfsprinzipien für Systemaufrufe und diskutieren deren spezifischen Vor-/Nachteile. • können grundlegende Mechanismen zur Speicherverwaltung implementieren • erschließen sich typische Probleme (Nebenläufigkeit, Compilerverhalten, Debuggen ohne dedizierte Hilfsmittel) und Fehlerquellen bei der hardwarenahen Softwareentwicklung. • sind in der Lage grundlegende Abstraktionen zur Adressraumvirtualisierung zu entwerfen. • können die notwendigen Voraussetzungen für Privilegientrennung benennen und diese auch technisch umsetzen • implementieren Techniken zur Latenzverbergung bei Speicheroperationen (Copy-On Write) • implementieren Mechanismen und Abstraktionen zur Interprozesskommunikation.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • können in Kleingruppen kooperativ arbeiten. • können die ihre Entwurfs- und Implementierungsentscheidungen kompakt präsentieren und argumentativ vertreten.		
<i>Selbstständigkeit</i>	Studierende, die das Modul erfolgreich abgeschlossen haben: • können sich komplexe Fehlerbilder durch methodisches Vorgehen schrittweise erschließen. • reflektieren ihre Entscheidungen kritisch und leiten geeignete Alternativen ab. • können offen und konstruktiv mit Schwachpunkten und Irrwegen umgehen. • können getroffene Fehlentscheidungen revidieren bzw. nehmen die entstehenden Kosten nur bewusst in Kauf. • können abstrakt gestellte Aufgaben zielgerichtet umsetzen.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	25 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht		

Lehrveranstaltung L2815: Betriebssystemtechnik	
Typ	Vorlesung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Christian Dietrich
Sprachen	DE
Zeitraum	WiSe
Inhalt	Schwerpunktthema der Veranstaltung ist die Verwaltung von Prozessadressräumen. Untersucht werden Verfahren und Techniken zur Trennung logischer Adressräume, zum Adressraumgrenzen überschreitenden Zugriff und zum Schutz von Prozessen. Vorgestellt wird die Implementierung von Systemaufrufen und seiten- wie auch segmentbasierte Techniken zur Abbildung logischer/virtueller Adressräume auf reale. Vor diesem Hintergrund werden verschiedene Betriebssystemarchitekturen verglichen und gängige Adressraummodelle von Betriebssystemen erläutert. Weitere Themen bildet die Interprozesskommunikation durch Nachrichtenversenden bei getrennten Adressräumen, aber auch die Nachbildung virtuell gemeinsamen Speichers auf Basis solcher Ansätze. Die Vorlesung vermittelt die notwendigen Kenntnisse um ein gegebenes Kleinstbetriebssystem um Speicherschutz und Privilegienisolation zu erweitern.
Literatur	

Lehrveranstaltung L2816: Betriebssystemtechnik	
Typ	Projekt-/problemorientierte Lehrveranstaltung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Christian Dietrich
Sprachen	DE
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1780: Massively Parallel Systems: Architecture and Programming				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Massiv parallele Systeme: Architektur und Programmierung (L2936)		Vorlesung	2	3
Massiv parallele Systeme: Architektur und Programmierung (L2937)		Projekt-/problembasierte Lehrveranstaltung	2	3
Modulverantwortlicher	Prof. Sohan Lal			
Zulassungsvoraussetzungen	None			
Empfohlene Vorkenntnisse	An introductory module on computer Engineering or computer architecture, good programming skills in C/C++.			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz				
Wissen	The course starts with parallel computers classification, multithreading, and covers the architecture of centralized and distributed shared-memory parallel systems, multiprocessor cache coherence, snooping / directory-based cache coherence protocols, implementation, and limitations. Next, students study interconnection networks and routing in parallel systems. To ensure the correctness of shared-memory multithreaded programs, independent of the speed of execution of their individual threads, the important topics of memory consistency and synchronization will be covered in detail. As a case study, the architecture of a few accelerators such as GPUs will also be discussed in detail. Besides understanding the architecture and organization of parallel systems, programming them is also very challenging. The course will also cover how to program massively parallel systems using API/libraries such as CUDA/OpenCL/MPI/OpenMP.			
Fertigkeiten	After completing this course, students will be able to understand the architecture and organization of parallel systems. They will be able to evaluate different design choices and make decisions while designing a parallel system. In addition, they will be able to program parallel systems (ranging from an embedded system to a supercomputer) using CUDA/OpenCL/MPI/OpenMP.			
Personale Kompetenzen				
Sozialkompetenz	The course will encourage students to work in small groups to solve complex problems, thus, inculcating the importance of teamwork.			
Selbstständigkeit	Today, parallel computers are present everywhere. Students will be able to not only program parallel computers independently, but also understand their underlying organization and architecture. This will further help to understand the performance issues of parallel applications and provide insights to improve them.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Verpflichtend Bonus Ja 20 %	Art der Studienleistung Fachtheoretisch- fachpraktische Studienleistung	Beschreibung	
Prüfung	Mündliche Prüfung			
Prüfungsdauer und -umfang	25 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht			

Lehrveranstaltung L2936: Massively Parallel Systems: Architecture and Programming	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sohan Lal
Sprachen	EN
Zeitraum	WiSe
Inhalt	Brief outline: - Parallel computers and their classification - Centralized and distributed shared-memory architectures: snooping vs directory-based cache coherence protocols, implementation, and limitations - Chip multiprocessors: software-based, block (coarse-grain), interleaved (fine-grain), simultaneous multithreading - Synchronization: high-level primitives and implementation, memory consistency models: sequential and weaker memory consistency models - Interconnection networks: topologies (direct and indirect networks) and routing techniques - Graphics Processing Units (GPUs) architecture and programming using CUDA/OpenCL - Parallel programming with message passing interface (MPI), OpenMP
Literatur	- Michel Dubois, Murali Annavaram, and Per Stenström, Parallel Computer Organization and Design (Book) - David A Patterson and John L. Hennessy, Computer Architecture: A Quantitative Approach, Elsevier (Book) - David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, Elsevier (Book)

Lehrveranstaltung L2937: Massively Parallel Systems: Architecture and Programming	
Typ	Projekt-/problem-basierte Lehrveranstaltung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sohan Lal
Sprachen	EN
Zeitraum	WiSe
Inhalt	There will be 3-4 assignments for project-based learning consisting of the following: • Implement and compare different cache coherence protocols using a simulator or a high-level, event-driven simulation interface such as SystemC • Programming massively parallel systems to solve computationally intensive problems such as password cracking using CUDA/OpenCL/MPI/OpenMP
Literatur	The following literature will be useful for project-based learning. The further required resources will be discussed during the course. • David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors, Elsevier (Book) • MPI Forum, https://www.mpi-forum.org/ • SystemC, https://www.accellera.org/community/systemc

Fachmodule der Vertiefung II. Intelligenz-Engineering

Modul M0633: Industrial Process Automation

Lehrveranstaltungen

Titel	Typ	SWS	LP
Prozessautomatisierungstechnik (L0344)	Vorlesung	2	3
Prozessautomatisierungstechnik (L0345)	Gruppenübung	2	3

Modulverantwortlicher	Prof. Alexander Schlaefer
------------------------------	---------------------------

Zulassungsvoraussetzungen	None
----------------------------------	------

Empfohlene Vorkenntnisse	mathematics and optimization methods principles of automata principles of algorithms and data structures programming skills
---------------------------------	--

Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht
---	---

Fachkompetenz	
<i>Wissen</i>	The students can evaluate and assess discrete event systems. They can evaluate properties of processes and explain methods for process analysis. The students can compare methods for process modelling and select an appropriate method for actual problems. They can discuss scheduling methods in the context of actual problems and give a detailed explanation of advantages and disadvantages of different programming methods. The students can relate process automation to methods from robotics and sensor systems as well as to recent topics like 'cyberphysical systems' and 'industry 4.0'.
<i>Fertigkeiten</i>	The students are able to develop and model processes and evaluate them accordingly. This involves taking into account optimal scheduling, understanding algorithmic complexity, and implementation using PLCs.
Personale Kompetenzen	
<i>Sozialkompetenz</i>	The students work in teams to solve problems.
<i>Selbstständigkeit</i>	The students can reflect their knowledge and document the results of their work.

Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56
----------------------------------	-------------------------------------

Leistungspunkte	6
------------------------	---

Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Nein 10 %	Übungsaufgaben	

Prüfung	Klausur
----------------	---------

Prüfungsdauer und -umfang	90 Minuten
----------------------------------	------------

Zuordnung zu folgenden Curricula	Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Regelungs- und Energiesystemtechnik: Wahlpflicht Flugzeug-Systemtechnik: Kernqualifikation: Wahlpflicht Flugzeug-Systemtechnik: Vertiefung Kabinensysteme: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Mechatronik: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Produktentwicklung und Produktion: Wahlpflicht Mechanical Engineering and Management: Vertiefung Mechatronik: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht Verfahrenstechnik: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht
---	---

Lehrveranstaltung L0344: Industrial Process Automation	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Schlaefer
Sprachen	EN
Zeitraum	WiSe
Inhalt	- foundations of problem solving and system modeling, discrete event systems - properties of processes, modeling using automata and Petri-nets - design considerations for processes (mutex, deadlock avoidance, liveness) - optimal scheduling for processes - optimal decisions when planning manufacturing systems, decisions under uncertainty - software design and software architectures for automation, PLCs
Literatur	J. Lunze: „Automatisierungstechnik“, Oldenbourg Verlag, 2012 Reisig: Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien; Vieweg+Teubner 2010 Hrúz, Zhou: Modeling and Control of Discrete-event Dynamic Systems; Springer 2007 Li, Zhou: Deadlock Resolution in Automated Manufacturing Systems, Springer 2009 Pinedo: Planning and Scheduling in Manufacturing and Services, Springer 2009

Lehrveranstaltung L0345: Industrial Process Automation	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Schlaefer
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0550: Digital Image Analysis			
Lehrveranstaltungen			
Titel Digitale Bildanalyse (L0126)	Typ Vorlesung	SWS 4	LP 6
Modulverantwortlicher	Prof. Rolf-Rainer Grigat		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	System theory of one-dimensional signals (convolution and correlation, sampling theory, interpolation and decimation, Fourier transform, linear time-invariant systems), linear algebra (Eigenvalue decomposition, SVD), basic stochastics and statistics (expectation values, influence of sample size, correlation and covariance, normal distribution and its parameters), basics of Matlab, basics in optics		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i>	Students can - Describe imaging processes - Depict the physics of sensorics - Explain linear and non-linear filtering of signals - Establish interdisciplinary connections in the subject area and arrange them in their context - Interpret effects of the most important classes of imaging sensors and displays using mathematical methods and physical models.		
<i>Fertigkeiten</i>	Students are able to - Use highly sophisticated methods and procedures of the subject area - Identify problems and develop and implement creative solutions. Students can solve simple arithmetical problems relating to the specification and design of image processing and image analysis systems. Students are able to assess different solution approaches in multidimensional decision-making areas. Students can undertake a prototypical analysis of processes in Matlab.		
Personale Kompetenzen <i>Sozialkompetenz</i>	k.A.		
<i>Selbstständigkeit</i>	Students can solve image analysis tasks independently using the relevant literature.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	60 Minuten, Umfang Vorlesung und Materialien im StudIP		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Nachrichten- und Kommunikationstechnik: Wahlpflicht Elektrotechnik: Vertiefung Medizintechnik: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Signalverarbeitung: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme, Schwerpunkt Software und Signalverarbeitung : Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Informationstechnologie: Wahlpflicht Mechatronik: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Microelectronics and Microsystems: Vertiefung Communication and Signal Processing: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht		

Lehrveranstaltung L0126: Digital Image Analysis	
Typ	Vorlesung
SWS	4
LP	6
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56
Dozenten	Prof. Rolf-Rainer Grigat
Sprachen	EN
Zeitraum	WiSe
Inhalt	• Image representation, definition of images and volume data sets, illumination, radiometry, multispectral imaging, reflectivities, shape from shading • Perception of luminance and color, color spaces and transforms, color matching functions, human visual system, color appearance models • imaging sensors (CMOS, CCD, HDR, X-ray, IR), sensor characterization(EMVA1288), lenses and optics • spatio-temporal sampling (interpolation, decimation, aliasing, leakage, moiré, flicker, apertures) • features (filters, edge detection, morphology, invariance, statistical features, texture) • optical flow (variational methods, quadratic optimization, Euler-Lagrange equations) • segmentation (distance, region growing, cluster analysis, active contours, level sets, energy minimization and graph cuts) • registration (distance and similarity, variational calculus, iterative closest points)
Literatur	Bredies/Lorenz, Mathematische Bildverarbeitung, Vieweg, 2011 Wedel/Cremers, Stereo Scene Flow for 3D Motion Analysis, Springer 2011 Handels, Medizinische Bildverarbeitung, Vieweg, 2000 Pratt, Digital Image Processing, Wiley, 2001 Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989

Modul M0629: Intelligent Autonomous Agents and Cognitive Robotics			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Intelligente Autonome Agenten und kognitive Robotik (L0341)	Vorlesung	2	4
Intelligente Autonome Agenten und kognitive Robotik (L0512)	Gruppenübung	2	2
Modulverantwortlicher	Rainer Marrone		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Vectors, matrices, Calculus		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Students can explain the agent abstraction, define intelligence in terms of rational behavior, and give details about agent design (goals, utilities, environments). They can describe the main features of environments. The notion of adversarial agent cooperation can be discussed in terms of decision problems and algorithms for solving these problems. For dealing with uncertainty in real-world scenarios, students can summarize how Bayesian networks can be employed as a knowledge representation and reasoning formalism in static and dynamic settings. In addition, students can define decision making procedures in simple and sequential settings, with and with complete access to the state of the environment. In this context, students can describe techniques for solving (partially observable) Markov decision problems, and they can recall techniques for measuring the value of information. Students can identify techniques for simultaneous localization and mapping, and can explain planning techniques for achieving desired states. Students can explain coordination problems and decision making in a multi-agent setting in term of different types of equilibria, social choice functions, voting protocol, and mechanism design techniques.		
<i>Fertigkeiten</i>	Students can select an appropriate agent architecture for concrete agent application scenarios. For simplified agent application students can derive decision trees and apply basic optimization techniques. For those applications they can also create Bayesian networks/dynamic Bayesian networks and apply bayesian reasoning for simple queries. Students can also name and apply different sampling techniques for simplified agent scenarios. For simple and complex decision making students can compute the best action or policies for concrete settings. In multi-agent situations students will apply techniques for finding different equilibria states,e.g., Nash equilibria. For multi-agent decision making students will apply different voting protocols and compare and explain the results.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Students are able to discuss their solutions to problems with others. They communicate in English		
<i>Selbstständigkeit</i>	Students are able of checking their understanding of complex concepts by solving variants of concrete problems		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Informationstechnologie: Wahlpflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mediziningenieurwesen: Vertiefung Künstliche Organe und Regenerative Medizin: Wahlpflicht Mediziningenieurwesen: Vertiefung Implantate und Endoprothesen: Wahlpflicht Mediziningenieurwesen: Vertiefung Medizin- und Regelungstechnik: Wahlpflicht Mediziningenieurwesen: Vertiefung Management und Administration: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht		

Lehrveranstaltung L0341: Intelligent Autonomous Agents and Cognitive Robotics	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Rainer Marrone
Sprachen	EN
Zeitraum	WiSe
Inhalt	• Definition of agents, rational behavior, goals, utilities, environment types • Adversarial agent cooperation: Agents with complete access to the state(s) of the environment, games, Minimax algorithm, alpha-beta pruning, elements of chance • Uncertainty: Motivation: agents with no direct access to the state(s) of the environment, probabilities, conditional probabilities, product rule, Bayes rule, full joint probability distribution, marginalization, summing out, answering queries, complexity, independence assumptions, naive Bayes, conditional independence assumptions • Bayesian networks: Syntax and semantics of Bayesian networks, answering queries revised (inference by enumeration), typical-case complexity, pragmatics: reasoning from effect (that can be perceived by an agent) to cause (that cannot be directly perceived). • Probabilistic reasoning over time: Environmental state may change even without the agent performing actions, dynamic Bayesian networks, Markov assumption, transition model, sensor model, inference problems: filtering, prediction, smoothing, most-likely explanation, special cases: hidden Markov models, Kalman filters, Exact inferences and approximations • Decision making under uncertainty: Simple decisions: utility theory, multivariate utility functions, dominance, decision networks, value of information Complex decisions: sequential decision problems, value iteration, policy iteration, MDPs Decision-theoretic agents: POMDPs, reduction to multidimensional continuous MDPs, dynamic decision networks • Simultaneous Localization and Mapping • Planning • Game theory (Golden Balls: Split or Share) Decisions with multiple agents, Nash equilibrium, Bayes-Nash equilibrium • Social Choice Voting protocols, preferences, paradoxes, Arrow's Theorem, • Mechanism Design Fundamentals, dominant strategy implementation, Revelation Principle, Gibbard-Satterthwaite Impossibility Theorem, Direct mechanisms, incentive compatibility, strategy-proofness, Vickrey-Groves-Clarke mechanisms, expected externality mechanisms, participation constraints, individual rationality, budget balancedness, bilateral trade, Myerson-Satterthwaite Theorem
Literatur	1. Artificial Intelligence: A Modern Approach (Third Edition), Stuart Russell, Peter Norvig, Prentice Hall, 2010, Chapters 2-5, 10-11, 13-17 2. Probabilistic Robotics, Thrun, S., Burgard, W., Fox, D. MIT Press 2005 3. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Yoav Shoham, Kevin Leyton-Brown, Cambridge University Press, 2009

Lehrveranstaltung L0512: Intelligent Autonomous Agents and Cognitive Robotics	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Rainer Marrone
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0630: Robotics and Navigation in Medicine			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Robotik und Navigation in der Medizin (L0335)	Vorlesung	2	3
Robotik und Navigation in der Medizin (L0338)	Projektseminar	2	2
Robotik und Navigation in der Medizin (L0336)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Alexander Schlaefer		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	- principles of math (algebra, analysis/calculus) - principles of programming, e.g., in Java or C++ - solid R or Matlab skills		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	The students can explain kinematics and tracking systems in clinical contexts and illustrate systems and their components in detail. Systems can be evaluated with respect to collision detection and safety and regulations. Students can assess typical systems regarding design and limitations.		
<i>Fertigkeiten</i>	The students are able to design and evaluate navigation systems and robotic systems for medical applications.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	The students discuss the results of other groups, provide helpful feedback and can incorporate feedback into their work.		
<i>Selbstständigkeit</i>	The students can reflect their knowledge and document the results of their work. They can present the results in an appropriate manner.		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Ja 10 %	Schriftliche Ausarbeitung	
	Ja 10 %	Referat	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Medizintechnik: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Elektrotechnik: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Verfahrenstechnik und Biotechnologie: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mediziningenieurwesen: Vertiefung Künstliche Organe und Regenerative Medizin: Wahlpflicht Mediziningenieurwesen: Vertiefung Implantate und Endoprothesen: Wahlpflicht Mediziningenieurwesen: Vertiefung Medizin- und Regelungstechnik: Wahlpflicht Mediziningenieurwesen: Vertiefung Management und Administration: Wahlpflicht Produktentwicklung, Werkstoffe und Produktion: Vertiefung Produktentwicklung: Wahlpflicht Produktentwicklung, Werkstoffe und Produktion: Vertiefung Produktion: Wahlpflicht Produktentwicklung, Werkstoffe und Produktion: Vertiefung Werkstoffe: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Bio- und Medizintechnik: Wahlpflicht		

Lehrveranstaltung L0335: Robotics and Navigation in Medicine	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Schlaefer
Sprachen	EN
Zeitraum	SoSe
Inhalt	- kinematics - calibration - tracking systems - navigation and image guidance - motion compensation The seminar extends and complements the contents of the lecture with respect to recent research results.
Literatur	Spong et al.: Robot Modeling and Control, 2005 Troccaz: Medical Robotics, 2012 Further literature will be given in the lecture.

Lehrveranstaltung L0338: Robotics and Navigation in Medicine	
Typ	Projektseminar
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Alexander Schlaefer
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L0336: Robotics and Navigation in Medicine	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Alexander Schlaefer
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1702: Process Imaging				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Prozessbildgebung (L2723)		Vorlesung	2	3
Prozessbildgebung (L2724)		Projekt-/problembasierte Lehrveranstaltung	2	3
Modulverantwortlicher	Prof. Alexander Penn			
Zulassungsvoraussetzungen	None			
Empfohlene Vorkenntnisse				
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>				
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Keine			
Prüfung	Klausur			
Prüfungsdauer und -umfang	120 min			
Zuordnung zu folgenden Curricula	Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht Bioverfahrenstechnik: Vertiefung B - Industrielle Bioverfahrenstechnik: Wahlpflicht Bioverfahrenstechnik: Vertiefung B - Industrielle Bioverfahrenstechnik: Wahlpflicht Bioverfahrenstechnik: Vertiefung C - Bioökonomische Verfahrenstechnik, Schwerpunkt Energie und Bioprozesstechnik: Wahlpflicht Bioverfahrenstechnik: Vertiefung C - Bioökonomische Verfahrenstechnik, Schwerpunkt Energie und Bioprozesstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Bioverfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Bioverfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Signalverarbeitung: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Verfahrenstechnik und Biotechnologie: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Umweltverfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Umweltverfahrenstechnik: Wahlpflicht Wasser- und Umweltingenieurwesen: Vertiefung Umwelt: Wahlpflicht Wasser- und Umweltingenieurwesen: Vertiefung Umwelt: Wahlpflicht Wasser- und Umweltingenieurwesen: Vertiefung Wasser: Wahlpflicht Wasser- und Umweltingenieurwesen: Vertiefung Wasser: Wahlpflicht			

Lehrveranstaltung L2723: Process Imaging	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Penn
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2724: Process Imaging	
Typ	Projekt-/problem-basierte Lehrveranstaltung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Penn, Dr. Stefan Benders
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Modul M0627: Machine Learning and Data Mining			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Maschinelles Lernen und Data Mining (L0340)	Vorlesung	2	4
Maschinelles Lernen und Data Mining (L0510)	Gruppenübung	2	2
Modulverantwortlicher	NN		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	• Calculus • Stochastics		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Students can explain the difference between instance-based and model-based learning approaches, and they can enumerate basic machine learning technique for each of the two basic approaches, either on the basis of static data, or on the basis of incrementally incoming data . For dealing with uncertainty, students can describe suitable representation formalisms, and they explain how axioms, features, parameters, or structures used in these formalisms can be learned automatically with different algorithms. Students are also able to sketch different clustering techniques. They depict how the performance of learned classifiers can be improved by ensemble learning, and they can summarize how this influences computational learning theory. Algorithms for reinforcement learning can also be explained by students.		
<i>Fertigkeiten</i>	Student derive decision trees and, in turn, propositional rule sets from simple and static data tables and are able to name and explain basic optimization techniques. They present and apply the basic idea of first-order inductive learning. Students apply the BME, MAP, ML, and EM algorithms for learning parameters of Bayesian networks and compare the different algorithms. They also know how to carry out Gaussian mixture learning. They can contrast kNN classifiers, neural networks, and support vector machines, and name their basic application areas and algorithmic properties. Students can describe basic clustering techniques and explain the basic components of those techniques. Students compare related machine learning techniques, e.g., k-means clustering and nearest neighbor classification. They can distinguish various ensemble learning techniques and compare the different goals of those techniques.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>			
<i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Internationales Wirtschaftsingenieurwesen: Vertiefung II. Informationstechnologie: Wahlpflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht Mechatronics: Vertiefung Systementwurf: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht		

Lehrveranstaltung L0340: Machine Learning and Data Mining	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Rainer Marrone
Sprachen	EN
Zeitraum	SoSe
Inhalt	• Decision trees • First-order inductive learning • Incremental learning: Version spaces • Uncertainty • Bayesian networks • Learning parameters of Bayesian networks BME, MAP, ML, EM algorithm • Learning structures of Bayesian networks • Gaussian Mixture Models • kNN classifier, neural network classifier, support vector machine (SVM) classifier • Clustering Distance measures, k-means clustering, nearest neighbor clustering • Kernel Density Estimation • Ensemble Learning • Reinforcement Learning • Computational Learning Theory
Literatur	1. Artificial Intelligence: A Modern Approach (Third Edition), Stuart Russel, Peter Norvig, Prentice Hall, 2010, Chapters 13, 14, 18-21 2. Machine Learning: A Probabilistic Perspective, Kevin Murphy, MIT Press 2012

Lehrveranstaltung L0510: Machine Learning and Data Mining	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Rainer Marrone
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1302: Angewandte Humanoide Robotik			
Lehrveranstaltungen			
Titel Angewandte Humanoide Robotik (L1794)	Typ Projekt-/problembasierte Lehrveranstaltung	SWS 6	LP 6
Modulverantwortlicher	Patrick Götttsch		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	• Objektorientierte Programmierung, Algorithmen und Datenstrukturen • Grundlagen der Regelungstechnik • Control systems theory and design • Mechanik		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> • Die Studierenden können Eigenschaften der humanoiden Robotik nennen und erläutern. • Die Studierenden können die grundlegenden Theorien, Zusammenhänge und Methoden der Vorwärts- & Rückwärtskinematik von humanoiden Robotersystemen erklären. • Die Studierenden können Regelkonzepte für verschiedene Aufgaben der Humanoiden Robotik anwenden. <i>Fertigkeiten</i> • Die Studierenden können die Modelle der Systeme der humanoiden Robotik in Matlab und C++ implementieren und diese Modelle für Bewegungen des Roboters oder andere Aufgaben nutzen. • Sie sind in der Lage die Modelle in Matlab für Simulationen zu nutzen und dann ggf. auch mit C++ Code auf dem realen Robotersystem zu testen. • Sie sind darüber hinaus in der Lage, für eine abstrakte Aufgabenstellung, für die es keine standardisierte Lösung gibt, Methoden auszuwählen, die zu gewünschten Ergebnissen führen. Personale Kompetenzen <i>Sozialkompetenz</i> • Die Studierenden können in fachlich gemischten Teams gemeinsame Lösungen entwickeln und diese vor anderen vertreten. • Sie sind in der Lage angemessenes Feedback zu geben und mit Rückmeldungen zu ihren eigenen Leistungen konstruktiv umzugehen. <i>Selbstständigkeit</i> • Die Studierenden sind in der Lage, die notwendigen Informationen aus den angegebenen Literaturquellen zu beschaffen und in den Kontext der Lehrveranstaltung zu setzen. • Sie können sich eigenständig Aufgaben definieren und geeignete Mittel zur Umsetzung einsetzen.			
Arbeitsaufwand in Stunden	Eigenstudium 96, Präsenzstudium 84		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Schriftliche Ausarbeitung		
Prüfungsdauer und -umfang	5-10 Seiten		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Regelungs- und Energiesystemtechnik: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Bio- und Medizintechnik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht		

Lehrveranstaltung L1794: Angewandte Humanoide Robotik	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	6
LP	6
Arbeitsaufwand in Stunden	Eigenstudium 96, Präsenzstudium 84
Dozenten	Patrick Götttsch
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	• Grundlagen der Kinematik • Grundlagen der statischen und dynamischen Stabilität humanoider Robotersysteme • Verknüpfung verschiedener Entwicklungsumgebungen (Matlab, C++, etc.) • Einarbeitung in die notwendigen Frameworks • Bearbeitung einer Projektaufgabe im Team • Präsentation und Demonstration von Zwischen- und Endergebnissen
Literatur	• B. Siciliano, O. Khatib. "Handbook of Robotics. Part A: Robotics Foundations", Springer (2008)

Modul M1249: Medizinische Bildgebung				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Medizinische Bildgebung (L1694)		Vorlesung	2	3
Medizinische Bildgebung (L1695)		Gruppenübung	2	3
Modulverantwortlicher	Prof. Tobias Knopp			
Zulassungsvoraussetzungen	Keine			
Empfohlene Vorkenntnisse	Grundkenntnisse in Linear Algebra, Numerik und Signalverarbeitung			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz				
<i>Wissen</i>	Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Rekonstruktionsverfahren für verschiedene tomographische Bildgebungsmodalitäten wie die Computertomographie und die Magnetresonanztomographie zu beschreiben. Sie kennen die nötigen Grundlagen aus den Bereichen der Signalverarbeitung und der inversen Probleme und kennen sowohl analytische als auch iterative Bildrekonstruktionsmethoden. Die Studierenden verfügen über vertiefende Kenntnisse über die Bildgebungsoperatoren der Computertomographie und die Magnetresonanztomographie.			
<i>Fertigkeiten</i>	Die Studierenden sind dazu in der Lage, Rekonstruktionsverfahren zu implementieren und diese anhand von tomographischen Messdaten zu testen. Sie können die rekonstruierten Bilder visualisieren und die Qualität ihrer Daten und Resultate und beurteilen. Zudem können die Studierenden die zeitliche Komplexität von Bildgebungsalgorithmen abschätzen.			
Personale Kompetenzen				
<i>Sozialkompetenz</i>	Die Studierenden können in sowohl selbstständig als auch in Teams an komplexen Problemen arbeiten. Sie können sich untereinander austauschen und ihre individuellen Stärken zur Lösung des Problems einbringen.			
<i>Selbstständigkeit</i>	Die Studierenden sind in der Lage ein komplexes Problem eigenständig zu untersuchen und einzuschätzen, welche Kompetenzen zur Lösung des Problems benötigt werden.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Keine			
Prüfung	Klausur			
Prüfungsdauer und -umfang	90 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Medizintechnik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Interdisciplinary Mathematics: Vertiefung III. Computational Methods in Biomedical Imaging: Pflicht Microelectronics and Microsystems: Vertiefung Communication and Signal Processing: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Bio- und Medizintechnik: Wahlpflicht			

Lehrveranstaltung L1694: Medizinische Bildgebung	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	• Überblick über bekannte Bildgebungsverfahren • Signalverarbeitung • Inverse Probleme • Computertomographie • Magnetresonanztomographie • Compressed Sensing • Magnetic-Particle-Imaging
Literatur	Bildgebende Verfahren in der Medizin; O. Dössel; Springer, Berlin, 2000 Bildgebende Systeme für die medizinische Diagnostik; H. Morneburg (Hrsg.); Publicis MCD, München, 1995 Introduction to the Mathematics of Medical Imaging; C. L. Epstein; Siam, Philadelphia, 2008 Medical Image Processing, Reconstruction and Restoration; J. Jan; Taylor and Francis, Boca Raton, 2006 Principles of Magnetic Resonance Imaging; Z.-P. Liang and P. C. Lauterbur; IEEE Press, New York, 1999

Lehrveranstaltung L1695: Medizinische Bildgebung	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0623: Intelligent Systems in Medicine			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Intelligente Systeme in der Medizin (L0331)	Vorlesung	2	3
Intelligente Systeme in der Medizin (L0334)	Projektseminar	2	2
Intelligente Systeme in der Medizin (L0333)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Alexander Schläefer		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	- principles of math (algebra, analysis/calculus) - principles of stochastics - principles of programming, Java/C++ and R/Matlab - advanced programming skills		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	The students are able to analyze and solve clinical treatment planning and decision support problems using methods for search, optimization, and planning. They are able to explain methods for classification and their respective advantages and disadvantages in clinical contexts. The students can compare different methods for representing medical knowledge. They can evaluate methods in the context of clinical data and explain challenges due to the clinical nature of the data and its acquisition and due to privacy and safety requirements.		
<i>Fertigkeiten</i>	The students can give reasons for selecting and adapting methods for classification, regression, and prediction. They can assess the methods based on actual patient data and evaluate the implemented methods.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	The students are able to grasp practical tasks in groups, develop solution strategies independently, define work processes and work on them collaboratively. The students can critically reflect on the results of other groups, make constructive suggestions for improvement and also incorporate them into their own work.		
<i>Selbstständigkeit</i>	The students can assess their level of knowledge and document their work results. They can critically evaluate the results achieved and present them in an appropriate argumentative manner to the other groups.		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Ja 10 %	Schriftliche Ausarbeitung	
	Ja 10 %	Referat	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Intelligenz-Engineering: Wahlpflicht Elektrotechnik: Vertiefung Medizintechnik: Wahlpflicht Interdisciplinary Mathematics: Vertiefung III. Computational Methods in Biomedical Imaging: Pflicht Mechatronik: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mediziningenieurwesen: Vertiefung Künstliche Organe und Regenerative Medizin: Wahlpflicht Mediziningenieurwesen: Vertiefung Implantate und Endoprothesen: Wahlpflicht Mediziningenieurwesen: Vertiefung Medizin- und Regelungstechnik: Wahlpflicht Mediziningenieurwesen: Vertiefung Management und Administration: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Bio- und Medizintechnik: Wahlpflicht		

Lehrveranstaltung L0331: Intelligent Systems in Medicine	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Schläefer
Sprachen	EN
Zeitraum	WiSe
Inhalt	- methods for search, optimization, planning, classification, regression and prediction in a clinical context - representation of medical knowledge - understanding challenges due to clinical and patient related data and data acquisition The students will work in groups to apply the methods introduced during the lecture using problem based learning.
Literatur	Russel & Norvig: Artificial Intelligence: a Modern Approach, 2012 Berner: Clinical Decision Support Systems: Theory and Practice, 2007 Greenes: Clinical Decision Support: The Road Ahead, 2007 Further literature will be given in the lecture

Lehrveranstaltung L0334: Intelligent Systems in Medicine	
Typ	Projektseminar
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Alexander Schläefer
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L0333: Intelligent Systems in Medicine	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Alexander Schläefer
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Fachmodule der Vertiefung III. Mathematik

Modul M0667: Algorithmische Algebra

Lehrveranstaltungen

Titel		Typ	SWS	LP
Algorithmische Algebra (L0422)		Vorlesung	3	5
Algorithmische Algebra (L0423)		Gruppenübung	1	1
Modulverantwortlicher	Dr. Prashant Batra			
Zulassungsvoraussetzungen	Keine			
Empfohlene Vorkenntnisse	Mathe I-III (Reelle Analysis, Rechnen in Vektorräumen, Vollst. Induktion) Diskrete Mathematik I (Gruppen, Ringe, Ideale, Körper; euklidischer Algorithmus)			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz	<i>Wissen</i> Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern: Smith-Normalform, Chinesischer Restsatz, Gitterpunktsätze, Ganzzahlige Lösung von Ungleichungssystemen. <i>Fertigkeiten</i> Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren. Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten, wie beispielsweise bei der Lösung multivariater Gleichungssysteme und in der Gitterpunkttheorie.			
Personale Kompetenzen				
<i>Sozialkompetenz</i> , <i>Selbstständigkeit</i>				
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56			
Leistungspunkte	6			
Studienleistung	Keine			
Prüfung	Mündliche Prüfung			
Prüfungsdauer und -umfang	30 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht			

Lehrveranstaltung L0422: Algorithmische Algebra

Typ	Vorlesung
SWS	3
LP	5
Arbeitsaufwand in Stunden	Eigenstudium 108, Präsenzstudium 42
Dozenten	Dr. Prashant Batra
Sprachen	DE
Zeitraum	WiSe
Inhalt	Erweiterter Euklidischer Algorithmus, Lösen der Bezout-Gleichung Teilen mit Rest in Ringen Schnelle Rechenalgorithmen (Konversion in Zahlformate, Schnelle Multiplikationen) Diskrete Fourier-Transformation in Ringe Rechnen mit modularen Resten, Lösen von Restsystemen (Chinesischer Restsatz), Lösbarkeit ganzzahliger Gleichungssysteme Linearisierung polynomialer Gleichungen - Matrizenansatz Sylvester-Matrix, Elimination Elimination in Ringen, Elimination mehrerer Veränderlicher Buchberger-Algorithmus, Gröbner-Basis Minkowskischer Gitterpunktsatz und Ganzzahlige Optimierung LLL-Algorithmus zum Auffinden 'kurzer' Vektoren in polynomialer Zeit

Literatur	von zur Gathen, Joachim; Gerhard, Jürgen Modern computer algebra. 3rd ed. (English) Zbl 1277.68002 Cambridge: Cambridge University Press (ISBN 978-1-107-03903-2/hbk; 978-1-139-85606-5/ebook). Yap, Chee Keng Fundamental problems of algorithmic algebra. (English) Zbl 0999.68261 Oxford: Oxford University Press. xvi, 511 p. \$ 87.00 (2000). Free download for students from author's website: http://cs.nyu.edu/yap/book/berlin/ Cox, David; Little, John; O'Shea, Donal Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra. 3rd ed. (English) Zbl 1118.13001 Undergraduate Texts in Mathematics. New York, NY: Springer (ISBN 978-0-387-35650-1/hbk; 978-0-387-35651-8/ebook). xv, 551 p. eBook: http://dx.doi.org/10.1007/978-0-387-35651-8 Concrete abstract algebra : from numbers to Gröbner bases / Niels Lauritzen Verfasser: Lauritzen, Niels Ausgabe: Reprinted with corr. Erschienen: Cambridge [u.a.] : Cambridge Univ. Press, 2006 Umfang: XIV, 240 S. : graph. Darst. Anmerkung: Includes bibliographical references and index ISBN: 0-521-82679-9, 978-0-521-82679-2 (hbk.) : GBP 55.00 0-521-53410-0, 978-0-521-53410-9 (pbk.) : USD 39.99 Koeppf, Wolfram Computer algebra. An algorithmic oriented introduction. (Computeralgebra. Eine algorithmisch orientierte Einführung.) (German) Zbl 1161.68881 Berlin: Springer (ISBN 3-540-29894-0/pbk). xiii, 515 p. springer eBook: http://dx.doi.org/10.1007/3-540-29895-9 Kaplan, Michael Computer algebra. (Computeralgebra.) (German) Zbl 1093.68148 Berlin: Springer (ISBN 3-540-21379-1/pbk). xii, 391 p. springer eBook: http://dx.doi.org/10.1007/b137968
------------------	---

Lehrveranstaltung L0423: Algorithmische Algebra	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dr. Prashant Batra
Sprachen	DE
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1428: Lineare und Nichtlineare Optimierung			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Lineare und Nichtlineare Optimierung (L2062)	Vorlesung	4	4
Lineare und Nichtlineare Optimierung (L2063)	Hörsaalübung	1	2
Modulverantwortlicher	Prof. Matthias Mnich		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	• Diskrete Algebraische Strukturen • Mathematik I • Graphentheorie und Optimierung		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>	• Studierende können die grundlegenden Begriffe der Linearen und Nichtlinearen Optimierung benennen und anhand von Beispielen erklären. • Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern. • Sie kennen Beweisstrategien und können diese wiedergeben. • Studierende können Aufgabenstellungen der Linearen und Nichtlinearen Optimierung mit Hilfe der kennengelernten Konzepte mathematisch modellieren und mit den erlernten Methoden lösen. • Studierende sind in der Lage, sich weitere einfache logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren. • Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten. • Studierende sind in der Lage, in heterogen zusammengestellten Teams (mit unterschiedlichem mathematischen Hintergrundwissen und aus unterschiedlichen Studiengängen) zusammenzuarbeiten und die Mathematik als gemeinsame Sprache zu entdecken und beherrschen. • Sie können sich dabei insbesondere gegenseitig neue Konzepte erklären und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen. • Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen. • Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume an schwierigen Problemstellungen zu arbeiten.		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung III. Mathematik: Wahlpflicht		

Lehrveranstaltung L2062: Lineare und Nichtlineare Optimierung	
Typ	Vorlesung
SWS	4
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 64, Präsenzstudium 56
Dozenten	Prof. Matthias Mnich
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	• Modellierung mit Hilfe von linearen Programmen • Lösung durch die graphische Methode • Algebraisches Hintergrundwissen • Konvexität • Polyeder • Simplex-Algorithmus • Degeneriertheit und Konvergenz • Dualität • Innere-Punkte Methoden • Quadratische Optimierung
Literatur	• A. Schrijver: Combinatorial Optimization: Polyhedra and Efficiency. Springer, 2003 • B. Korte and T. Vygen: Combinatorial Optimization: Theory and Algorithms. Springer, 2018 • T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013

Lehrveranstaltung L2063: Lineare und Nichtlineare Optimierung	
Typ	Hörsaalübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Matthias Mnich
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0716: Hierarchische Algorithmen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Hierarchische Algorithmen (L0585)	Vorlesung	2	3
Hierarchische Algorithmen (L0586)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Sabine Le Borne		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Mathematik I, II, III für Ingenieurstudierende (deutsch oder englisch) oder Analysis & Lineare Algebra I + II sowie Analysis III für Technomathematiker - Programmierenkenntnisse in C		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Studierende können - Vertreter hierarchischer Algorithmen benennen und ihre grundlegenden Merkmale herausstellen, - Konstruktionstechniken hierarchischer Algorithmen erklären, - Aspekte der effizienten Implementierung von hierarchischen Algorithmen diskutieren. <i>Fertigkeiten</i> Studierende sind in der Lage, - die in der Vorlesung behandelten hierarchischen Algorithmen zu implementieren, - den Speicherbedarf und die Rechenzeitkomplexität der Algorithmen zu analysieren, - die Algorithmen an Problemstellungen unterschiedlicher Anwendungen anzupassen und somit problemadaptierte Varianten zu entwickeln. Personale Kompetenzen <i>Sozialkompetenz</i> Studierende können - in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten, sich theoretische Grundlagen erklären sowie bei praktischen Implementierungsaspekten der Algorithmen unterstützen. <i>Selbstständigkeit</i> Studierende sind fähig, - selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen, - mit ausreichender Ausdauer komplexe Problemstellungen über längere Zeiträume zu bearbeiten, - ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.		
Arbeitsaufwand in Stunden			
Leistungspunkte			
Studienleistung			
Prüfung			
Prüfungsdauer und -umfang	20 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Simulationstechnik: Wahlpflicht		

Lehrveranstaltung L0585: Hierarchische Algorithmen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	- Niedrigrangmatrizen - Separable Entwicklungen - Hierarchische Matrixpartitionen - Hierarchische Matrizen - Formatierte Matrixoperationen - Anwendungen - weitere Themen (z.B. H2-Matrizen, Matrixfunktionen, Tensorprodukte)
Literatur	W. Hackbusch: Hierarchische Matrizen: Algorithmen und Analysis

Lehrveranstaltung L0586: Hierarchische Algorithmen	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1405: Randomisierte Algorithmen und Zufällige Graphen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Randomisierte Algorithmen und Zufällige Graphen (L2010)	Vorlesung	2	3
Randomisierte Algorithmen und Zufällige Graphen (L2011)	Hörsaalübung	2	3
Modulverantwortlicher	Prof. Anusch Taraz		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Algorithmen und Datenstrukturen - Mathematik I und II - Stochastik - Graphentheorie		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> - Students can describe basic concepts in the area of Randomized Algorithms and Random Graphs such as random walks, tail bounds, fingerprinting and algebraic techniques, first and second moment methods, and various random graph models. They are able to explain them using appropriate examples. - Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. - They know proof strategies and can apply them. <i>Fertigkeiten</i> - Students can model problems with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. - Students are able to explore and verify further logical connections between the concepts studied in the course. - For a given problem, the students can develop and execute a suitable technique, and are able to critically evaluate the results. Personale Kompetenzen <i>Sozialkompetenz</i> - Students are able to work together in teams. They are capable to establish a common language. - In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers. <i>Selbstständigkeit</i> - Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. - Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung III. Mathematik: Wahlpflicht		

Lehrveranstaltung L2010: Randomisierte Algorithmen und Zufällige Graphen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Anusch Taraz, Prof. Volker Turau
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Randomized Algorithms: • introduction and recalling basic tools from probability • randomized search • random walks • text search with fingerprinting • parallel and distributed algorithms • online algorithms Random Graphs: • typical properties • first and second moment method • tail bounds • thresholds and phase transitions • probabilistic method • models for complex networks
Literatur	• Motwani, Raghavan: Randomized Algorithms • Worsch: Randomisierte Algorithmen • Dietzfelbinger: Randomisierte Algorithmen • Bollobas: Random Graphs • Alon, Spencer: The Probabilistic Method • Frieze, Karonski: Random Graphs • van der Hofstad: Random Graphs and Complex Networks

Lehrveranstaltung L2011: Randomisierte Algorithmen und Zufällige Graphen	
Typ	Hörsaalübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Anusch Taraz, Prof. Volker Turau
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0714: Numerik gewöhnlicher Differentialgleichungen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Numerik gewöhnlicher Differentialgleichungen (L0576)	Vorlesung	2	3
Numerik gewöhnlicher Differentialgleichungen (L0582)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Daniel Ruprecht		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Mathematik I, II, III für Ingenieurstudierende (deutsch oder englisch) oder Analysis & Lineare Algebra I + II sowie Analysis III für Technomathematiker - MATLAB Grundkenntnisse		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Studierende können - numerische Verfahren zur Lösung gewöhnlicher Differentialgleichungen benennen und deren Kernideen erläutern, - Konvergenzaussagen (inklusive der an das zugrundeliegende Problem gestellten Voraussetzungen) zu den behandelten numerischen Verfahren wiedergeben, - Aspekte der praktischen Durchführung numerischer Verfahren erklären. - Wählen Sie die entsprechende numerische Methode für konkrete Probleme, implementieren die numerischen Algorithmen effizient und interpretieren die numerischen Ergebnisse <i>Fertigkeiten</i> Studierende sind in der Lage, - numerische Methoden zur Lösung gewöhnlicher Differentialgleichungen in MATLAB zu implementieren, anzuwenden und zu vergleichen, - das Konvergenzverhalten numerischen Methoden in Abhängigkeit vom gestellten Problem und des verwendeten Lösungsalgorithmus zu begründen, - zu gegebener Problemstellung einen geeigneten Lösungsansatz zu entwickeln, gegebenenfalls durch Zusammensetzen mehrerer Algorithmen, diesen durchzuführen und die Ergebnisse kritisch auszuwerten. Personale Kompetenzen <i>Sozialkompetenz</i> Studierende können - in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten, sich theoretische Grundlagen erklären sowie bei praktischen Implementierungsaspekten der Algorithmen unterstützen. <i>Selbstständigkeit</i> Studierende sind fähig, - selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen, - ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.		
Arbeitsaufwand in Stunden			
Leistungspunkte			
Studienleistung			
Prüfung			
Prüfungsdauer und -umfang			
Zuordnung zu folgenden Curricula	Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Chemical and Bioprocess Engineering: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht Computer Science: Vertiefung III. Mathematik: Wahlpflicht Elektrotechnik: Vertiefung Regelungs- und Energiesystemtechnik: Wahlpflicht Energietechnik: Kernqualifikation: Wahlpflicht Flugzeug-Systemtechnik: Kernqualifikation: Wahlpflicht Interdisciplinary Mathematics: Vertiefung II. Numerical - Modelling Training: Pflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Theoretischer Maschinenbau: Kernqualifikation: Pflicht Verfahrenstechnik: Vertiefung Chemische Verfahrenstechnik: Wahlpflicht Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht		

Lehrveranstaltung L0576: Numerik gewöhnlicher Differentialgleichungen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Numerische Verfahren für Anfangswertprobleme • Einschrittverfahren • Mehrschrittverfahren • Steife Probleme • Differentiell-algebraische Gleichungen vom Index 1 Numerische Verfahren für Randwertaufgaben • Mehrzielmethode • Differenzenverfahren • Variationsmethoden
Literatur	• E. Hairer, S. Noersett, G. Wanner: Solving Ordinary Differential Equations I: Nonstiff Problems • E. Hairer, G. Wanner: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems

Lehrveranstaltung L0582: Numerik gewöhnlicher Differentialgleichungen	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1668: Probability Theory			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Wahrscheinlichkeitstheorie (L2643)	Vorlesung	3	4
Wahrscheinlichkeitstheorie (L2644)	Gruppenübung	1	2
Modulverantwortlicher	Prof. Matthias Schulte		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Familiarity with the basic concepts of probability		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> - Students can name the basic concepts in probability theory. They are able to explain them using appropriate examples. - Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples. - They know proof strategies and can reproduce them. <i>Fertigkeiten</i> - Students can model problems from probability theory with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods. - Students are able to explore and verify further logical connections between the concepts studied in the course. - For a given problem, the students can develop and execute a suitable technique, and are able to critically evaluate the results. Personale Kompetenzen <i>Sozialkompetenz</i> - Students are able to work together (e.g. on their regular home work) and to present their results appropriately (e.g. during exercise class). - In doing so, they can communicate new concepts according to the needs of their cooperating partners. Moreover, they can design examples to check and deepen the understanding of their peers. <i>Selbstständigkeit</i> - Students are capable of checking their understanding of complex concepts on their own. They can specify open questions precisely and know where to get help in solving them. - Students can put their knowledge in relation to the contents of other lectures. - Students have developed sufficient persistence to be able to work for longer periods in a goal-oriented manner on hard problems.			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Interdisciplinary Mathematics: Vertiefung II. Numerical - Modelling Training: Pflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht		

Lehrveranstaltung L2643: Probability Theory	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Matthias Schulte
Sprachen	EN
Zeitraum	SoSe
Inhalt	• Measure and probability spaces • Integration and expectation • Types of stochastic convergence • Law of large numbers • Central limit theorem • Radon-Nikodym theorem • Conditional expectation • Martingales • Markov chains • Poisson processes
Literatur	H. Bauer, Probability theory and elements of measure theory, second edition, Academic Press, 1981. A. Klenke, Probability Theory: A Comprehensive Course, second edition, Springer, 2014. G. F. Lawler, Introduction to Stochastic Processes, second edition, Chapman & Hall/CRC, 2006. A. N. Shiryaev, Probability, second edition, Springer, 1996.

Lehrveranstaltung L2644: Probability Theory	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Matthias Schulte
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0711: Numerische Mathematik II			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Numerische Mathematik II (L0568)	Vorlesung	2	3
Numerische Mathematik II (L0569)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Sabine Le Borne		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Numerische Mathematik I - Python-Kenntnisse		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Studierende können - weiterführende numerische Verfahren zur Interpolation, Approximation, Integration, Lösung von Eigenwertaufgaben, Lösung von Eigenwertproblemen und nichtlinearen Nullstellenproblemen benennen und deren Kernideen erläutern, - Konvergenzbeweise skizzieren, - Aspekte der praktischen Durchführung numerischer Verfahren im Hinblick auf Rechenzeit und Speicherbedarf erklären. - Konvergenzaussagen zu den numerischen Methoden wiedergeben Studierende sind in der Lage, - vertiefende numerische Methoden in Python zu implementieren, anzuwenden und zu vergleichen, - das Konvergenzverhalten numerischen Methoden in Abhängigkeit vom gestellten Problem und des verwendeten Lösungsalgorithmus zu begründen und auf verwandte Problemstellungen zu übertragen - zu gegebener Problemstellung einen geeigneten Lösungsansatz zu entwickeln, gegebenenfalls durch Zusammensetzen mehrerer Algorithmen, diesen durchzuführen und die Ergebnisse kritisch auszuwerten.		
Wissen			
Fertigkeiten			
Personale Kompetenzen			
Sozialkompetenz	Studierende können in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten, sich theoretische Grundlagen erklären sowie bei praktischen Implementierungsaspekten der Algorithmen unterstützen.		
Selbstständigkeit	Studierende sind fähig, - selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen, - ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	25 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung III. Mathematik: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Theoretischer Maschinenbau: Kernqualifikation: Wahlpflicht		

Lehrveranstaltung L0568: Numerische Mathematik II	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	1. Fehler und Stabilität: Begriffe und Abschätzungen 2. Rationale Interpolation und Approximation 3. Mehrdimensionale Interpolation (RBF) und Approximation (neuronale Netze) 4. Quadratur: Gauß-Quadratur, Orthogonalpolynome 5. Lineare Systeme: Perturbationstheorie von Zerlegungen, strukturierte Matrizen 6. Eigenwertaufgaben: LR-, QD-, QR-Algorithmus 7. Nichtlineare Gleichungssysteme: Newton- und Quasi-Newton-Verfahren, Liniensuche (optional) 8. Krylovraum-Verfahren: Arnoldi-, Lanczos-Verfahren (optional)
Literatur	- Skript - Stoer/Bulirsch: Numerische Mathematik 1, Springer - Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer

Lehrveranstaltung L0569: Numerische Mathematik II	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0881: Mathematische Bildverarbeitung			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Mathematische Bildverarbeitung (L0991)	Vorlesung	3	4
Mathematische Bildverarbeitung (L0992)	Gruppenübung	1	2
Modulverantwortlicher	Prof. Marko Lindner		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	• Analysis: partielle Ableitungen, Gradient, Richtungsableitung • Lineare Algebra: Eigenwerte, lineares Ausgleichsproblem		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Die Studierenden können • Klassen von Diffusionsgleichungen charakterisieren und vergleichen • elementare Methoden der Bildverarbeitung erklären • Methoden zur Segmentierung und Registrierung erläutern • funktionalanalytische Grundlagen skizzieren und gegenüberstellen <i>Fertigkeiten</i> Die Studierenden können • elementare Methoden der Bildverarbeitung implementieren und anwenden • moderne Methoden der Bildverarbeitung erklären und anwenden Personale Kompetenzen <i>Sozialkompetenz</i> Studierende können in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten und sich theoretische Grundlagen erklären. <i>Selbstständigkeit</i> • Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen. • Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume an schwierigen Problemstellungen zu arbeiten.		
Arbeitsaufwand in Stunden			
Leistungspunkte			
Studienleistung			
Prüfung			
Prüfungsdauer und -umfang			
Zuordnung zu folgenden Curricula	Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht Computer Science: Vertiefung III. Mathematik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung III. Mathematik: Wahlpflicht Interdisciplinary Mathematics: Vertiefung III. Computational Methods in Biomedical Imaging: Pflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht Mechatronics: Vertiefung Systementwurf: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht		

Lehrveranstaltung L0991: Mathematische Bildverarbeitung	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Marko Lindner
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	• Elementare Methoden der Bildverarbeitung • Glättungsfilter • Grundlagen der Diffusions- bzw. Wärmeleitgleichung • Variationsformulierungen in der Bildverarbeitung • Kantenerkennung • Entfaltung • Inpainting • Segmentierung • Registrierung
Literatur	Bredies/Lorenz: Mathematische Bildverarbeitung

Lehrveranstaltung L0992: Mathematische Bildverarbeitung	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Marko Lindner
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1552: Fortgeschrittenes maschinelles Lernen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Fortgeschrittenes maschinelles Lernen (L2322)	Vorlesung	2	3
Fortgeschrittenes maschinelles Lernen (L2323)	Gruppenübung	2	3
Modulverantwortlicher	Dr. Jens-Peter Zemke		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	1. Mathematik I-III 2. Numerische Mathematik 1/ Numerik 3. Programmierkenntnisse, bestenfalls in Python		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Studierende können die mathematischen Grundlagen verschiedener neuronaler Netze benennen, wiedergeben, neuronale Netze klassifizieren und hinsichtlich der Schwierigkeiten bewerten. Studierende können neuronale Netze implementieren, verstehen und gezielt sowie an die Problemstellung angepasst anwenden. Studierende können - in kleinen Gruppen Lösungen erarbeiten und dokumentieren; - in Gruppen Ideen weiterentwickeln und auf anderen Kontext übertragen; - im Team eine Software-Bibliothek entwickeln, aufbauen und weiterentwickeln. Studierende sind fähig - den Aufwand und Umfang selbst definierter Aufgaben korrekt einzuschätzen; - selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen; - sich eigenständig Aufgaben zum Test und zum Ausbau der Verfahren auszudenken; - ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.		
<i>Wissen</i>			
<i>Fertigkeiten</i>			
Personale Kompetenzen			
<i>Sozialkompetenz</i>			
<i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	25 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung III. Mathematik: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Robotik und Informatik: Wahlpflicht		

Lehrveranstaltung L2322: Fortgeschrittenes maschinelles Lernen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Dr. Jens-Peter Zemke
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	1. Grundlagen: Analogie, Aufbau neuronaler Netze, universelle Approximationseigenschaft, NP-Vollständigkeit 2. Feedforward-Netze: Backpropagation, Varianten des stochastischen Gradientenverfahrens 3. Deep Learning: Probleme und Lösungsstrategien 4. Deep Belief Networks: Energie-basierte Modelle, Contrastive Divergence 5. Faltungsnetze: Idee, Aufbau, FFT und Algorithmen von Winograd, Implementationsdetails 6. Rekurrente Netze: Idee, dynamische Systeme, Training, LSTM 7. Residuale Netze: Idee, Verbindung zu neuronalen ODEs 8. Standardbibliotheken: Tensorflow, Keras, PyTorch 9. Neue Trends
Literatur	1. Skript 2. Online-Werke: http://neuralnetworksanddeeplearning.com/ https://www.deeplearningbook.org/

Lehrveranstaltung L2323: Fortgeschrittenes maschinelles Lernen	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Dr. Jens-Peter Zemke
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1020: Numerik partieller Differentialgleichungen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Numerik partieller Differentialgleichungen (L1247)	Vorlesung	2	3
Numerik partieller Differentialgleichungen (L1248)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Daniel Ruprecht		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	- Mathematik I - IV (für Ingenieurstudierende) oder Analysis & Lineare Algebra I + II für Technomathematiker - Numerische Mathematik 1 - Numerik gewöhnlicher Differentialgleichungen		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i>	- Die Studierenden können partielle Differentialgleichungen den drei Grundtypen zuordnen. - Sie kennen für jeden Typ die passenden numerischen Zugänge. - Sie kennen das Konvergenzverhalten dieser Verfahren.		
<i>Fertigkeiten</i>	Die Studierenden sind in der Lage, zu gegebenen partiellen Differentialgleichungsproblemen numerische Lösungsansätze zu formulieren, theoretische Konvergenzaussagen zu treffen sowie diese Ansätze in der Praxis durchzuführen, d.h. zu implementieren und zu testen.		
Personale Kompetenzen <i>Sozialkompetenz</i>	Studierende können in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten und sich theoretische Grundlagen erklären.		
<i>Selbstständigkeit</i>	- Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen. - Studierende haben eine genügend hohe Ausdauer entwickelt, um auch über längere Zeiträume an schwierigen Problemstellungen zu arbeiten.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	30 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung III. Mathematik: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Theoretischer Maschinenbau: Vertiefung Simulationstechnik: Wahlpflicht		

Lehrveranstaltung L1247: Numerik partieller Differentialgleichungen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Elementare Theorie und Numerik Partielle Differentialgleichungen: - Typen partieller Differentialgleichungen - wohlgestellte Probleme - Finite Differenzen - Finite Volumen - Anwendungen
Literatur	Dale R. Durran: Numerical Methods for Fluid Dynamics. Randall J. LeVeque: Numerical Methods for Conservation Laws.

Lehrveranstaltung L1248: Numerik partieller Differentialgleichungen	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Fachmodule der Vertiefung IV. Fachspezifische Fokussierung

Die drei Technischen Ergänzungsfächer dienen der weiteren Vertiefung der im Wahlpflichtbereich gewählten Module.

Modul M1565: Technischer Ergänzungskurs I für CSMS			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Modulverantwortlicher	Prof. Karl-Heinz Zimmermann		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Abhängig von der Wahl der Lehrveranstaltungen		
Leistungspunkte	6		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung IV. Fachspezifische Fokussierung: Wahlpflicht		

Modul M1566: Technischer Ergänzungskurs II für CSMS			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Modulverantwortlicher	Prof. Karl-Heinz Zimmermann		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i> <i>Fertigkeiten</i> Personale Kompetenzen <i>Sozialkompetenz</i> <i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Abhängig von der Wahl der Lehrveranstaltungen		
Leistungspunkte	6		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung IV. Fachspezifische Fokussierung: Wahlpflicht		

Modul M1564: Hauptseminare Informatik und Kommunikationstechnik			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Hauptseminar Informatik und Kommunikationstechnik I (L2352)	Seminar	2	3
Hauptseminar Informatik und Kommunikationstechnik II (L2429)	Seminar	2	3
Modulverantwortlicher	Dozenten des SD E		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Grundlegende Module aus der Informatik und Mathematik auf Masterebene.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<i>Wissen</i> Die Studierenden können • ein spezifisches Thema der Informatik erklären, • komplexe Sachverhalte beschreiben, • unterschiedliche Standpunkte darlegen und kritisch bewerten. <i>Fertigkeiten</i> Die Studierenden können • sich in einer begrenzten Zeit in ein spezifisches Thema der Informatik einarbeiten, • eine Literaturrecherche durchführen und die Quellen richtig zitieren und angeben, • selbstständig einen Vortrag ausarbeiten und vor ausgewählten Publikum halten, • den Vortrag in einem Abstract zusammenfassen, • im Rahmen der Diskussion Fachfragen beantworten. Personale Kompetenzen <i>Sozialkompetenz</i> Die Studierenden sind in der Lage, • ein Thema für eine bestimmte Zielgruppe aufzuarbeiten und darzustellen, • mit der Betreuerin bzw. dem Betreuer das Thema sowie Inhalt und Aufbau des Vortrages zu diskutieren, • einzelne Aspekte aus dem Themengebiet mit den Zuhörerinnen und Zuhörern durchzusprechen, • als Vortragende bzw. Vortragender auf die Fragen der Zuhörerinnen und Zuhörer einzugehen. <i>Selbstständigkeit</i> Die Studierenden werden die Lage versetzt, • eigenständig Aufgaben zu definieren, • notwendiges Wissen zu erschließen, • geeignete Hilfsmittel einzusetzen, • unter Anleitung der Betreuerin bzw. des Betreuers den Arbeitsstand kritisch zu überprüfen.		
Arbeitsaufwand in Stunden			
Leistungspunkte			
Studienleistung			
Prüfung			
Prüfungsdauer und -umfang			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung IV. Fachspezifische Fokussierung: Wahlpflicht Information and Communication Systems: Vertiefung Kommunikationssysteme: Wahlpflicht Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme: Wahlpflicht		

Lehrveranstaltung L2352: Hauptseminar Informatik und Kommunikationstechnik I	
Typ	Seminar
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2429: Hauptseminar Informatik und Kommunikationstechnik II	
Typ	Seminar
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	
Literatur	

Thesis

Modul M-002: Masterarbeit			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Modulverantwortlicher	Professoren der TUHH		
Zulassungsvoraussetzungen	Laut ASPO § 21 (1): Es müssen mindestens 60 Leistungspunkte im Studiengang erworben worden sein. Über Ausnahmen entscheidet der Prüfungsausschuss.		
Empfohlene Vorkenntnisse	keine		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz <i>Wissen</i>	- Die Studierenden können das Spezialwissen (Fakten, Theorien und Methoden) ihres Studienfaches sicher zur Bearbeitung fachlicher Fragestellungen einsetzen. - Die Studierenden können in einem oder mehreren Spezialbereichen ihres Faches die relevanten Ansätze und Terminologien in der Tiefe erklären, aktuelle Entwicklungen beschreiben und kritisch Stellung beziehen. - Die Studierenden können eine eigene Forschungsaufgabe in ihrem Fachgebiet verorten, den Forschungsstand erheben und kritisch einschätzen.		
Fertigkeiten	- Die Studierenden sind in der Lage, für die jeweilige fachliche Problemstellung geeignete Methoden auszuwählen, anzuwenden und ggf. weiterzuentwickeln. - Die Studierenden sind in der Lage, im Studium erworbenes Wissen und erlernte Methoden auch auf komplexe und/oder unvollständig definierte Problemstellungen lösungsorientiert anzuwenden. - Die Studierenden können in ihrem Fachgebiet neue wissenschaftliche Erkenntnisse erarbeiten und diese kritisch beurteilen.		
Personale Kompetenzen <i>Sozialkompetenz</i>	Studierende können - eine wissenschaftliche Fragestellung für ein Fachpublikum sowohl schriftlich als auch mündlich strukturiert, verständlich und sachlich richtig darstellen. - in einer Fachdiskussion Fragen fachkundig und zugleich adressatengerecht beantworten und dabei eigene Einschätzungen überzeugend vertreten.		
<i>Selbstständigkeit</i>	Studierende sind fähig, - ein eigenes Projekt in Arbeitspakete zu strukturieren und abzuarbeiten. - sich in ein teilweise unbekanntes Arbeitsgebiet des Studiengangs vertieft einzuarbeiten und dafür benötigte Informationen zu erschließen. - Techniken des wissenschaftlichen Arbeitens umfassend in einer eigenen Forschungsarbeit anzuwenden.		
Arbeitsaufwand in Stunden	Eigenstudium 900, Präsenzstudium 0		
Leistungspunkte	30		
Studienleistung	Keine		
Prüfung	Abschlussarbeit		
Prüfungsdauer und -umfang	laut ASPO		
Zuordnung zu folgenden Curricula	Bauingenieurwesen: Abschlussarbeit: Pflicht Bioverfahrenstechnik: Abschlussarbeit: Pflicht Chemical and Bioprocess Engineering: Abschlussarbeit: Pflicht Computer Science: Abschlussarbeit: Pflicht Elektrotechnik: Abschlussarbeit: Pflicht Energietechnik: Abschlussarbeit: Pflicht Environmental Engineering: Abschlussarbeit: Pflicht Flugzeug-Systemtechnik: Abschlussarbeit: Pflicht Global Innovation Management: Abschlussarbeit: Pflicht Informatik-Ingenieurwesen: Abschlussarbeit: Pflicht Information and Communication Systems: Abschlussarbeit: Pflicht Interdisciplinary Mathematics: Abschlussarbeit: Pflicht International Production Management: Abschlussarbeit: Pflicht Internationales Wirtschaftsingenieurwesen: Abschlussarbeit: Pflicht Joint European Master in Environmental Studies - Cities and Sustainability: Abschlussarbeit: Pflicht Logistik, Infrastruktur und Mobilität: Abschlussarbeit: Pflicht		

	Materialwissenschaft: Abschlussarbeit: Pflicht
	Mechanical Engineering and Management: Abschlussarbeit: Pflicht
	Mechatronics: Abschlussarbeit: Pflicht
	Mediziningenieurwesen: Abschlussarbeit: Pflicht
	Microelectronics and Microsystems: Abschlussarbeit: Pflicht
	Produktentwicklung, Werkstoffe und Produktion: Abschlussarbeit: Pflicht
	Regenerative Energien: Abschlussarbeit: Pflicht
	Schiffbau und Meerestechnik: Abschlussarbeit: Pflicht
	Ship and Offshore Technology: Abschlussarbeit: Pflicht
	Teilstudiengang Lehramt Metalltechnik: Abschlussarbeit: Pflicht
	Theoretischer Maschinenbau: Abschlussarbeit: Pflicht
	Verfahrenstechnik: Abschlussarbeit: Pflicht
	Wasser- und Umweltingenieurwesen: Abschlussarbeit: Pflicht
	Zulassungs- und Sachverständigenwesen in der Luftfahrt: Abschlussarbeit: Pflicht