

Modulhandbuch

Bachelor of Science (B.Sc.)

Computer Science

Kohorte: Wintersemester 2021

Stand: 18. September 2021

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Studiengangsbeschreibung	3
Fachmodule der Kernqualifikation	5
Modul M0561: Diskrete Algebraische Strukturen	5
Modul M0731: Functional Programming	7
Modul M0577: Nichttechnische Angebote im Bachelor	9
Modul M1436: Prozedurale Programmierung für Informatiker	11
Modul M1728: Mathematics I (EN)	12
Modul M1729: Mathematics II (EN)	14
Modul M0624: Automata Theory and Formal Languages	16
Modul M0829: Grundlagen der Betriebswirtschaftslehre	18
Modul M1432: Programmierparadigmen	21
Modul M1732: Mathematics III (EN)	23
Modul M0834: Computernetworks and Internet Security	25
Modul M0730: Technische Informatik	27
Modul M1423: Algorithmen und Datenstrukturen	29
Modul M0727: Stochastik	31
Modul M0732: Software Engineering	33
Modul M0852: Graphentheorie und Optimierung	35
Modul M0562: Berechenbarkeit und Komplexität	37
Modul M0873: Software-Fachpraktikum	39
Modul M1578: Seminare Informatik	40
Modul M1620: Ethik in der Informationstechnologie	41
Fachmodule der Vertiefung I. Computer- und Software-Engineering	43
Modul M0625: Databases	43
Modul M0971: Betriebssysteme	45
Modul M1586: Wissenschaftliche Programmierung	46
Modul M0791: Rechnerarchitektur	48
Modul M0953: Introduction to Information Security	50
Modul M0803: Embedded Systems	52
Modul M0754: Compiler Construction	54
Modul M1300: Software Development	56
Fachmodule der Vertiefung II. Mathematik und Ingenieurwissenschaften	58
Modul M0662: Numerical Mathematics I	58
Modul M1730: Mathematics IV (EN)	60
Modul M0651: Rechnergestützte Geometrie	62
Modul M0941: Kombinatorische Strukturen und Algorithmen	65
Modul M1242: Quantenmechanik für Studierende der Ingenieurwissenschaften	67
Modul M1592: Statistik	69
Modul M0634: Einführung in Medizintechnische Systeme	71
Modul M0668: Algebraische Methoden in der Regelungstechnik	73
Modul M1269: Labor Cyber-Physical Systems	75
Modul M0715: Solvers for Sparse Linear Systems	76
Modul M0672: Signale und Systeme	78
Fachmodule der Vertiefung III. Fachspezifische Fokussierung	81
Modul M1562: Technischer Ergänzungskurs I für CSBS	81
Modul M1568: Technischer Ergänzungskurs II für CSBS	82
Thesis	83
Modul M-001: Bachelorarbeit	83

Studiengangsbeschreibung

Inhalt

Die Informatik ist neben Biotechnologie, Medizintechnik und Nanotechnologie die Schlüsseltechnologie des 21. Jahrhunderts. Sie hat sich zu einer Triebfeder des technologischen Fortschrittes entwickelt, weil alle Berufszweige mit Informationsaspekten durchdrungen sind und immer neue Anwendungsfelder in der Informations- und Kommunikationstechnik erschlossen werden. Deshalb brauchen wir heute und morgen Informatikerinnen und Informatiker, die informationsverarbeitende Systeme qualifiziert und verantwortungsbewusst entwerfen, analysieren und an gegebene Einsatzbedingungen anpassen.

Die Informatik ist eng mit der Mathematik und der Elektrotechnik/Elektronik verbunden, ist aber auch als eine Basis- und Querschnittsdisziplin zu verstehen, die sich sowohl mit technischen als auch mit organisatorischen Problemen bei der Entwicklung und Anwendung informationsverarbeitender Systeme beschäftigt. Sie erforscht die grundsätzlichen Verfahrensweisen der Informationsverarbeitung und die allgemeinen Methoden der Anwendung solcher Verfahren in den verschiedensten Bereichen. Sie geht durch Abstraktion und Modellbildung sowohl über die konkreten technischen Realisierungen informationsverarbeitender Systeme als auch über die Besonderheiten spezieller Anwendungen hinaus und gelangt zur Formulierung allgemeiner Gesetzmäßigkeiten. Daraus entwickelt sie Standardlösungen für die Aufgaben der Praxis, z.B. bei der Bewältigung großer Daten- und Informationsmengen und der Steuerung komplexer Produktionsabläufe (Quelle: studienwahl.de, 05/2015).

Der Bachelorstudiengang Computer Science bietet ein wissenschaftlich fundiertes, grundlagenorientiertes Studium. Auf der Basis eines breiten und in ausgewählten Teilgebieten vertieften fachlichen Wissens werden die analytischen, kreativen und konstruktiven Fähigkeiten zur Konzipierung von informationsverarbeitenden Systemen entwickelt und gefördert. Vor allem wird die Fähigkeit zur Realisierung und Implementierung von programmierbaren Systemen erworben. Durch die Bearbeitung von vielfältigen Problemen aus verschiedenen Anwendungsbereichen entwickeln die Studierenden insgesamt eine sinnvolle Mischung aus praktischen und wissenschaftlichen Fähigkeiten. Schlüsselqualifikationen wie Teamarbeit und Präsentationstechniken werden gezielt vermittelt.

Die Informatik unterliegt schnellen Innovationen, weshalb besonderer Wert auf zukunftsfestes Wissen gelegt wird. Damit sollen die Studierenden in die Lage versetzt werden, auch die künftigen Entwicklungen der Informatik selbstständig und auf hohem Niveau in ihre berufliche Praxis und in ihren persönlichen Horizont zu integrieren. Aus diesem Grund hat der Bachelorstudiengang eine wissenschaftliche und methodenorientierte Grundausrichtung. Erworben werden vor allem gründliche Kenntnisse in Informatik und der dazu gehörigen Mathematik sowie Betriebswirtschaftslehre.

Berufliche Perspektiven

Der Bachelorstudiengang Computer Science bereitet die Absolventen und Absolventinnen sowohl auf eine berufliche Tätigkeit im IT-Sektor als auch auf ein aufbauendes Master-Studium vor. Die Absolventen und Absolventinnen werden in die Lage versetzt, komplexe IT-Lösungen zu entwerfen und technisch umzusetzen. Ferner werden methodische Grundlagen erworben, um sich stets an neue berufliche Entwicklungen und Innovationen anzupassen. Daher sollten die Absolventen und Absolventinnen in nahezu allen Branchen eine verantwortungsvolle Tätigkeit finden können.

Lernziele

Das Bachelorstudium Computer Science soll die Studierenden sowohl auf eine berufliche Tätigkeit als auch auf ein einschlägiges Master-Studium vorbereiten. Das hierfür notwendige methodische Grundlagenwissen wird im Rahmen des Studiums erworben. Die Lernergebnisse des Studiengangs werden durch ein Zusammenspiel von grundlegenden und weiterführenden Modulen aus Informatik, Mathematik und Betriebswirtschaftslehre erreicht. 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 Bachelorstudiengang Computer Science auf folgenden Gebieten erworben:

1. Die Absolventen und Absolventinnen kennen grundlegende Methoden und Verfahren zur mathematischen Modellbildung in der Informatik, wie etwa algebraisch spezifizierte abstrakte Datentypen, Automatenmodelle, formale Grammatiken, Modelle der Berechenbarkeit, Komplexitätsmodelle, graphentheoretische Netzwerke, Differentialgleichungen, stochastische Prozesse und dynamische Systeme im Sinne der Systemtheorie. Sie können diese beschreiben und vergleichen.
2. Die Absolventen und Absolventinnen kennen fundamentale Methoden und Verfahren zur Lösung oder Approximation von algorithmischen Entscheidungs- und Optimierungsproblemen, wie etwa automatische Differentiation, direkte erschöpfende Suche via Backtracking, Gradientenbasierte Verfahren, graphentheoretische Algorithmen, Heuristiken, lineare (ganzzahlige) Programmierung, Testen von Hypothesen, Theorem-Beweiser, sowie deren Analyse hinsichtlich Komplexität, Konvergenz und Güte. Sie sind in der Lage, diese zu skizzieren und zu diskutieren.
3. Die Absolventen und Absolventinnen kennen die Grundlagen des Software-Entwurfes und können hierbei auf gängige prozedurale, objektorientierte, funktionale und logikbasierte Programmiersprachen sowie grundlegende Datenstrukturen und Algorithmen zurückgreifen. Sie sind vertraut mit dem Betrieb von Software-Systemen unter Berücksichtigung der Organisation und Verarbeitung großer Datenmengen, der Verwaltung der zur Verfügung stehenden Betriebsmittel und einem verteilten Arrangement von Daten und Algorithmen.
4. Die Absolventen und Absolventinnen kennen den Aufbau, den Betrieb und die Organisation von Rechenanlagen und sie wissen, wie Algorithmen auf dem von-Neumann-Rechner oder einem Mikroprozessor ausgeführt werden. Sie wissen ferner, wie Hardware-Bausteine programmiertechnisch beschrieben und simuliert werden können und sie können die Einbettung eines Strukturmodells in einen technischen Rahmen skizzieren und implementieren.
5. Die Absolventen und Absolventinnen kennen eine Reihe von Anwendungsfällen valider mathematischer Modelle in der Informatik, wie etwa Algorithmen in Netzwerken, diskrete und schnelle Fourier-Transformation, Public-Key-Infrastrukturen sowie Sortier- und Suchverfahren.

Fertigkeiten

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

1. Die Absolventen und Absolventinnen sind im Stande, Instanzen formaler Modelle in der Informatik anhand einfacher Modellierungsansätze zu entwickeln, ihre Berechenbarkeit und Komplexität einzuschätzen und sie anhand geeigneter Programmierwerkzeuge umzusetzen.
2. Die Absolventen und Absolventinnen sind in der Lage, Instanzen von algorithmischen Entscheidungs- und Optimierungsproblemen unter Einsatz des Erlernten optimal oder näherungsweise zu lösen und die Lösungen zu analysieren.
3. Die Absolventen und Absolventinnen können Software-Komponenten in komplexere Softwaresysteme unter Benutzung der im Studium erarbeiteten Methoden integrieren und testen.
4. Die Absolventen und Absolventinnen werden in die Lage versetzt, Mikroprozessoren zu programmieren und Strukturbeschreibungen von einfachen Hardware-Bausteinen zu entwickeln, zu simulieren und zu bewerten.
5. Die Absolventen und Absolventinnen können vertraute Anwendungsfälle valider mathematischer Modelle aus der Informatik unter Verwendung einschlägiger Werkzeuge umsetzen und die Lösungen evaluieren.

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 in einem fachlich homogenen Team organisieren, spezifische Teilaufgaben übernehmen und den eigenen Beitrag reflektieren.
2. Die Absolventen und Absolventinnen sind in der Lage, sich in ein fachlich heterogenes Team einzugliedern, gemeinsame Lösungen zu erarbeiten und diese vor anderen zu vertreten.

Kompetenz zum selbständigen Arbeiten

Personale Kompetenzen umfassen neben der Kompetenz zum selbständigen Handeln auch System- und Lösungskompetenzen, allgemeinen Problemstellung auf spezifische Teilprobleme abzubilden sowie die Auswahl und das Beherrschen geeigneter Methoden und Verfahren zur Problemlösung.

1. Die Absolventen und Absolventinnen können sich selbständig ein eng umrissenes Teilgebiet der Informatik erschließen und die Ergebnisse im Rahmen eines kurzen Vortrages mit fortschrittlichen Präsentationstechniken oder eines mehrseitigen Aufsatzes detailliert zusammenfassen.
2. Die Absolventen und Absolventinnen sind in der Lage, fachlich eingegrenzte Teilprojekte unter Verwendung des im Studium Erlernten in einem komplexeren IT-Entwicklungsprojekt eigenverantwortlich zu bearbeiten.

Studiengangsstruktur

Das Curriculum des Bachelorstudienganges Computer Science ist wie folgt gegliedert:

- Kernqualifikation: 21 Module, 132 Leistungspunkte (LP), 1. - 6. Semester
 - Fachmodule aus dem Grundlagenbereich von Informatik und Mathematik (114 LP)
 - Software-Fachpraktikum (6 LP)
 - Grundlagen der Betriebswirtschaftslehre (6 LP)
 - Nichttechnische Ergänzungsfächer (6 LP)
- Vertiefung: 6 Module, 36 LP, 5. und 6. Semester
 - Richtungen: Computermathematik, Computer- und Software-Engineering
- Bachelorarbeit: 12 LP, 6. Semester

Damit ergibt sich ein Gesamtaufwand von 180 LP.

Der Studienplan ist mit einem Mobilitätsfenster versehen, so dass das fünfte Semester unter Umständen im Ausland absolviert werden kann.

Fachmodule der Kernqualifikation

Im Studium der Informatik werden zunächst die Grundlagen der praktischen, theoretischen und technischen Informatik sowie der dazu benötigten Mathematik gelehrt. An der TUHH wird großer Wert auf eine breite methodenorientierte Ausbildung gelegt.

Im Einzelnen werden folgende Lehrinhalte vermittelt:

- Praktische Informatik
 - Programmier-Paradigmen (funktional, imperativ, objektorientiert)
 - Algorithmentechnik
 - Datenstrukturen
 - Softwareentwicklung
- Theoretische Informatik
 - Automatentheorie und Formale Sprachen
 - Berechenbarkeit und Komplexitätstheorie
 - Graphentheorie und Optimierung
 - Informationssicherheit
- Technische Informatik
 - Digitaltechnik
 - Rechnerstrukturen
 - Betriebssysteme
 - Rechnernetze
- Mathematik
 - Diskrete Mathematik
 - Lineare Algebra
 - Analysis
 - Stochastik
 - Klassische Systemtheorie

Abgerundet wird das Kernangebot durch Veranstaltungen, welche die persönlichen, sozialen und methodischen Kompetenzen (Soft-Skills) erhöhen sollen:

- Einführung in die Betriebswirtschaftslehre
- Studium Generale (3 Veranstaltungen frei wählbar aus dem Angebot)
- Seminare Informatik und Mathematik (3 Seminare frei wählbar aus dem Angebot)
- Software-Fachpraktikum (sechswöchig, wird in der Regel in der Industrie absolviert)

Modul M0561: Diskrete Algebraische Strukturen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Diskrete Algebraische Strukturen (L0164)	Vorlesung	2	3
Diskrete Algebraische Strukturen (L0165)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Karl-Heinz Zimmermann		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Abiturkenntnisse in Mathematik.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Wissen: Die Studierenden kennen <ul style="list-style-type: none"> • zahlentheoretische und funktionsbasierte Modelle der Kryptographie sowie Grundlagen der linearen Codes; • den Aufbau und Struktur von Restklassenringen (Euklidische Ringe) und endlichen Körpern; • den Aufbau und die Struktur von Unter-, Summen- und Faktorstrukturen in algebraischen Gebilden sowie Homomorphismen zwischen diesen Strukturen; • den Aufbau und die Abzählung von elementaren kombinatorischen Strukturen; • die wichtigsten Beweiskonzepte der modernen Mathematik; • den Aufbau der höheren Mathematik basierend auf mathematischer Logik und Mengenlehre; • grundlegende Aspekte des Einsatzes von mathematischer Software (Computeralgebrasystem Maple) zur Lösung von algebraischen oder kombinatorischen Aufgabenstellungen. 		
<i>Fertigkeiten</i>	Fertigkeiten: Die Studierenden können <ul style="list-style-type: none"> • in Restklassenringen (Euklidischen Ringen) rechnen; • Unter-, Summen- und Faktorstrukturen in algebraischen Gebilden aufstellen und in ihnen rechnen sowie algebraische Strukturen durch Homomorphismen aufeinander beziehen; • elementar-kombinatorische Strukturen identifizieren und abzählen; • die Sprache der Mathematik, basierend auf Mathematischer Logik und Mengenlehre, dienstbar verwenden; • einfache, im Kontext stehende mathematische Aussagen beweisen; • einschlägige mathematische Software (Computeralgebrasystem Maple) zielgerichtet einsetzen. 		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische 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 Fachbüchern 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	Klausur
Prüfungsdauer und -umfang	120 min
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Orientierungsstudium: Kernqualifikation: Wahlpflicht

Lehrveranstaltung L0164: Diskrete Algebraische Strukturen	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Karl-Heinz Zimmermann
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L0165: Diskrete Algebraische Strukturen	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Karl-Heinz Zimmermann
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0731: Functional Programming			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Funktionales Programmieren (L0624)	Vorlesung	2	2
Funktionales Programmieren (L0625)	Hörsaalübung	2	2
Funktionales Programmieren (L0626)	Gruppenübung	2	2
Modulverantwortlicher	Prof. Sibylle Schupp		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Discrete mathematics at high-school level		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Students apply the principles, constructs, and simple design techniques of functional programming. They demonstrate their ability to read Haskell programs and to explain Haskell syntax as well as Haskell's read-eval-print loop. They interpret warnings and find errors in programs. They apply the fundamental data structures, data types, and type constructors. They employ strategies for unit tests of functions and simple proof techniques for partial and total correctness. They distinguish laziness from other evaluation strategies.		
<i>Wissen</i>			
<i>Fertigkeiten</i>	Students break a natural-language description down in parts amenable to a formal specification and develop a functional program in a structured way. They assess different language constructs, make conscious selections both at specification and implementations level, and justify their choice. They analyze given programs and rewrite them in a controlled way. They design and implement unit tests and can assess the quality of their tests. They argue for the correctness of their program.		
Personale Kompetenzen	Students practice peer programming with varying peers. They explain problems and solutions to their peer. They defend their programs orally. They communicate in English.		
<i>Sozialkompetenz</i>			
<i>Selbstständigkeit</i>	In programming labs, students learn under supervision (a.k.a. "Betreutes Programmieren") the mechanics of programming. In exercises, they develop solutions individually and independently, and receive feedback.		
Arbeitsaufwand in Stunden	Eigenstudium 96, Präsenzstudium 84		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Ja 15 %	Übungsaufgaben	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Wahlpflicht Engineering Science: Vertiefung Mechatronics: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0624: Functional Programming	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions • Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions • Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type • Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps) • Modules • Interactive Programming • Lazy Evaluation, Call-by-Value, Strictness • Design Recipes • Testing (axiom-based, invariant-based, against reference implementation) • Reasoning about Programs (equation-based, inductive) • Idioms of Functional Programming • Haskell Syntax and Semantics
Literatur	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Lehrveranstaltung L0625: Functional Programming	
Typ	Hörsaalübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions • Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions • Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type • Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps) • Modules • Interactive Programming • Lazy Evaluation, Call-by-Value, Strictness • Design Recipes • Testing (axiom-based, invariant-based, against reference implementation) • Reasoning about Programs (equation-based, inductive) • Idioms of Functional Programming • Haskell Syntax and Semantics
Literatur	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Lehrveranstaltung L0626: Functional Programming	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Functions, Currying, Recursive Functions, Polymorphic Functions, Higher-Order Functions • Conditional Expressions, Guarded Expressions, Pattern Matching, Lambda Expressions • Types (simple, composite), Type Classes, Recursive Types, Algebraic Data Type • Type Constructors: Tuples, Lists, Trees, Associative Lists (Dictionaries, Maps) • Modules • Interactive Programming • Lazy Evaluation, Call-by-Value, Strictness • Design Recipes • Testing (axiom-based, invariant-based, against reference implementation) • Reasoning about Programs (equation-based, inductive) • Idioms of Functional Programming • Haskell Syntax and Semantics
Literatur	Graham Hutton, Programming in Haskell, Cambridge University Press 2007.

Modul M0577: Nichttechnische Angebote im Bachelor	
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>	<p>Die Nichttechnischen Angebote (NTA)</p> <p>vermitteln 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.</p> <p>Die Lehrarchitektur</p> <p>besteht aus einem studiengangübergreifenden Pflichtstudienangebot. Durch dieses zentral konzipierte Lehrangebot wird die Profilierung der TUHH Ausbildung auch im Nichttechnischen Bereich gewährleistet.</p> <p>Die Lernarchitektur erfordert und übt eigenverantwortliche Bildungsplanung in Hinblick auf den individuellen Kompetenzaufbau ein und stellt dazu Orientierungswissen zu thematischen Schwerpunkten von Veranstaltungen bereit.</p> <p>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 Auslandsemester zu fördern, wird jedoch von einer Studienfixierung in konkreten Fachsemestern abgesehen.</p> <p>Die Lehr-Lern-Arrangements</p> <p>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.</p> <p>Die Lehrbereiche</p> <p>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.</p> <p>Das Kompetenzniveau</p> <p>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.</p> <p>Fachkompetenz (Wissen)</p> <p>Die Studierenden können</p> <ul style="list-style-type: none"> • ausgewählte Spezialgebiete innerhalb der jeweiligen nichttechnischen Mutterdisziplinen verorten, • 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 nichttechnischen Bereich ist).
<i>Fertigkeiten</i>	<p>Die Studierenden können in ausgewählten Teilbereichen</p> <ul style="list-style-type: none"> • grundlegende Methoden der genannten Wissenschaftsdisziplinen anwenden. • technische Phänomene, Modelle, Theorien usw. aus der Perspektive einer anderen, oben erwähnten Fachdisziplin befragen. • einfache 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>	<p>Die Studierenden sind fähig ,</p> <ul style="list-style-type: none"> • in unterschiedlichem Ausmaß kooperativ zu lernen

<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> • 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) . <p>Die Studierenden sind in ausgewählten Bereichen in der Lage,</p> <ul style="list-style-type: none"> • 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 M1436: Prozedurale Programmierung für Informatiker			
Lehrveranstaltungen			
Titel		Typ	SWS
Prozedurale Programmierung für Informatiker (L2163)		Vorlesung	1
Prozedurale Programmierung für Informatiker (L2164)		Hörsaalübung	1
Prozedurale Programmierung für Informatiker (L2165)		Laborpraktikum	2
			LP
			2
			1
			3
Modulverantwortlicher	NN		
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	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Technomathematik: Kernqualifikation: Pflicht		

Lehrveranstaltung L2163: Prozedurale Programmierung für Informatiker	
Typ	Vorlesung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Siegfried Rump
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L2164: Prozedurale Programmierung für Informatiker	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L2165: Prozedurale Programmierung für Informatiker	
Typ	Laborpraktikum
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	
Literatur	

Modul M1728: Mathematics I (EN)			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Analysis I (EN) (L2771)	Vorlesung	2	2
Analysis I (EN) (L2772)	Hörsaalübung	1	1
Analysis I (EN) (L2773)	Gruppenübung	1	1
Lineare Algebra I (EN) (L2774)	Vorlesung	2	2
Lineare Algebra I (EN) (L2775)	Hörsaalübung	1	1
Lineare Algebra I (EN) (L2776)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Daniel Ruprecht		
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 128, Präsenzstudium 112		
Leistungspunkte	8		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Engineering Science: Kernqualifikation: Pflicht		

Lehrveranstaltung L2771: Analysis I (EN)	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L2772: Analysis I (EN)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2773: Analysis I (EN)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2774: Linear Algebra I (EN)	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L2775: Linear Algebra I (EN)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2776: Linear Algebra I (EN)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1729: Mathematics II (EN)			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Analysis II (EN) (L2777)	Vorlesung	2	2
Analysis II (EN) (L2778)	Hörsaalübung	1	1
Analysis II (EN) (L2779)	Gruppenübung	1	1
Lineare Algebra II (EN) (L2780)	Vorlesung	2	2
Lineare Algebra II (EN) (L2781)	Hörsaalübung	1	1
Lineare Algebra II (EN) (L2782)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Daniel Ruprecht		
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 128, Präsenzstudium 112		
Leistungspunkte	8		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Engineering Science: Kernqualifikation: Pflicht		

Lehrveranstaltung L2777: Analysis II (English)	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2778: Analysis II (English)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2779: Analysis II (English)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2780: Linear Algebra II (English)	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2781: Linear Algebra II (English)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2782: Linear Algebra II (English)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Daniel Ruprecht
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0624: Automata Theory and Formal Languages			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Automatentheorie und Formale Sprachen (L0332)	Vorlesung	2	4
Automatentheorie und Formale Sprachen (L0507)	Gruppenübung	2	2
Modulverantwortlicher	Prof. Tobias Knopp		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Participating students should be able to - specify algorithms for simple data structures (such as, e.g., arrays) to solve computational problems - apply propositional logic and predicate logic for specifying and understanding mathematical proofs - apply the knowledge and skills taught in the module Discrete Algebraic Structures		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Students can explain syntax, semantics, and decision problems of propositional logic, and they are able to give algorithms for solving decision problems. Students can show correspondences to Boolean algebra. Students can describe which application problems are hard to represent with propositional logic, and therefore, the students can motivate predicate logic, and define syntax, semantics, and decision problems for this representation formalism. Students can explain unification and resolution for solving the predicate logic SAT decision problem. Students can also describe syntax, semantics, and decision problems for various kinds of temporal logic, and identify their application areas. The participants of the course can define various kinds of finite automata and can identify relationships to logic and formal grammars. The spectrum that students can explain ranges from deterministic and nondeterministic finite automata and pushdown automata to Turing machines. Students can name those formalism for which nondeterminism is more expressive than determinism. They are also able to demonstrate which decision problems require which expressivity, and, in addition, students can transform decision problems w.r.t. one formalism into decision problems w.r.t. other formalisms. They understand that some formalisms easily induce algorithms whereas others are best suited for specifying systems and their properties. Students can describe the relationships between formalisms such as logic, automata, or grammars.</p> <p><i>Fertigkeiten</i> Students can apply propositional logic as well as predicate logic resolution to a given set of formulas. Students analyze application problems in order to derive propositional logic, predicate logic, or temporal logic formulas to represent them. They can evaluate which formalism is best suited for a particular application problem, and they can demonstrate the application of algorithms for decision problems to specific formulas. Students can also transform nondeterministic automata into deterministic ones, or derive grammars from automata and vice versa. They can show how parsers work, and they can apply algorithms for the language emptiness problem in case of infinite words.</p>		
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 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Engineering Science: Vertiefung Mechatronics: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Orientierungsstudium: Kernqualifikation: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0332: Automata Theory and Formal Languages	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ol style="list-style-type: none"> 1. Propositional logic, Boolean algebra, propositional resolution, SAT-2KNF 2. Predicate logic, unification, predicate logic resolution 3. Temporal Logics (LTL, CTL) 4. Deterministic finite automata, definition and construction 5. Regular languages, closure properties, word problem, string matching 6. Nondeterministic automata: Rabin-Scott transformation of nondeterministic into deterministic automata 7. Epsilon automata, minimization of automata, elimination of e-edges, uniqueness of the minimal automaton (modulo renaming of states) 8. Myhill-Nerode Theorem: Correctness of the minimization procedure, equivalence classes of strings induced by automata 9. Pumping Lemma for regular languages: provision of a tool which, in some cases, can be used to show that a finite automaton principally cannot be expressive enough to solve a word problem for some given language 10. Regular expressions vs. finite automata: Equivalence of formalisms, systematic transformation of representations, reductions 11. Pushdown automata and context-free grammars: Definition of pushdown automata, definition of context-free grammars, derivations, parse trees, ambiguities, pumping lemma for context-free grammars, transformation of formalisms (from pushdown automata to context-free grammars and back) 12. Chomsky normal form 13. CYK algorithm for deciding the word problem for context-free grammars 14. Deterministic pushdown automata 15. Deterministic vs. nondeterministic pushdown automata: Application for parsing, LL(k) or LR(k) grammars and parsers vs. deterministic pushdown automata, compiler compiler 16. Regular grammars 17. Outlook: Turing machines and linear bounded automata vs general and context-sensitive grammars 18. Chomsky hierarchy 19. Mealy- and Moore automata: Automata with output (w/o accepting states), infinite state sequences, automata networks 20. Omega automata: Automata for infinite input words, Büchi automata, representation of state transition systems, verification w.r.t. temporal logic specifications (in particular LTL) 21. LTL safety conditions and model checking with Büchi automata, relationships between automata and logic 22. Fixed points, propositional mu-calculus 23. Characterization of regular languages by monadic second-order logic (MSO)
Literatur	<ol style="list-style-type: none"> 1. Logik für Informatiker Uwe Schöning, Spektrum, 5. Aufl. 2. Logik für Informatiker Martin Kreuzer, Stefan Kühling, Pearson Studium, 2006 3. Grundkurs Theoretische Informatik, Gottfried Vossen, Kurt-Ulrich Witt, Vieweg-Verlag, 2010. 4. Principles of Model Checking, Christel Baier, Joost-Pieter Katoen, The MIT Press, 2007

Lehrveranstaltung L0507: Automata Theory and Formal Languages	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0829: Grundlagen der Betriebswirtschaftslehre			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Betriebswirtschaftliche Übung (L0882)	Gruppenübung	2	3
Grundlagen der Betriebswirtschaftslehre (L0880)	Vorlesung	3	3
Modulverantwortlicher	Prof. Christoph Ihl		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Schulkenntnisse in Mathematik und Wirtschaft		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Die Studierenden können...		
<i>Wissen</i>	<ul style="list-style-type: none"> grundlegende Begriffe und Kategorien aus dem Bereich Wirtschaft und Management benennen und erklären grundlegende Aspekte wettbewerblchen Unternehmertums beschreiben (Betrieb und Unternehmung, betrieblicher Zielbildungsprozess) wesentliche betriebliche Funktionen erläutern, insb. Funktionen der Wertschöpfungskette (z.B. Produktion und Beschaffung, Innovationsmanagement, Absatz und Marketing) sowie Querschnittsfunktionen (z.B. Organisation, Personalmanagement, Supply Chain Management, Informationsmanagement) und die wesentlichen Aspekte von Entrepreneurship-Projekten benennen Grundlagen der Unternehmensplanung (Entscheidungstheorie, Planung und Kontrolle) wie auch spezielle Planungsaufgaben (z.B. Projektplanung, Investition und Finanzierung) erläutern Grundlagen des Rechnungswesens erklären (Buchführung, Bilanzierung, Kostenrechnung, Controlling) 		
<i>Fertigkeiten</i>	Die Studierenden können		
	<ul style="list-style-type: none"> Unternehmensziele definieren und in ein Zielsystem einordnen sowie Zielsysteme strukturieren Organisations- und Personalstrukturen von Unternehmen analysieren Methoden für Entscheidungsprobleme unter mehrfacher Zielsetzung, unter Ungewissheit sowie unter Risiko zur Lösung von entsprechenden Problemen anwenden Produktions- und Beschaffungssysteme sowie betriebliche Informationssysteme analysieren und einordnen Einfache preispolitische und weitere Instrumente des Marketing analysieren und anwenden Grundlegende Methoden der Finanzmathematik auf Investitions- und Finanzierungsprobleme anwenden Die Grundlagen der Buchhaltung, Bilanzierung, Kostenrechnung und des Controlling erläutern und Methoden aus diesen Bereichen auf einfache Problemstellungen anwenden. 		
Personale Kompetenzen	Die Studierenden sind in der Lage		
<i>Sozialkompetenz</i>	<ul style="list-style-type: none"> sich im Team zu organisieren und ein Projekt aus dem Bereich Entrepreneurship gemeinsam zu bearbeiten und einen Projektbericht zu erstellen erfolgreich problemlösungsorientiert zu kommunizieren respektvoll und erfolgreich zusammenzuarbeiten 		
<i>Selbstständigkeit</i>	Die Studierenden sind in der Lage		
	<ul style="list-style-type: none"> Ein Projekt in einem Team zu bearbeiten und einer Lösung zuzuführen unter Anleitung einen Projektbericht zu verfassen 		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Fachtheoretisch-fachpraktische Arbeit		
Prüfungsdauer und -umfang	mehrere schriftliche Leistungen über das Semester verteilt		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Kernqualifikation: Pflicht Bau- und Umweltingenieurwesen: Vertiefung Bauingenieurwesen: Wahlpflicht Bau- und Umweltingenieurwesen: Vertiefung Wasser und Umwelt: Wahlpflicht Bau- und Umweltingenieurwesen: Vertiefung Verkehr und Mobilität: Wahlpflicht Bioverfahrenstechnik: Kernqualifikation: Pflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Elektrotechnik: Kernqualifikation: Pflicht Energie- und Umwelttechnik: Kernqualifikation: Pflicht General Engineering Science (7 Semester): Vertiefung Elektrotechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Bauingenieurwesen: Pflicht General Engineering Science (7 Semester): Vertiefung Bioverfahrenstechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Energie- und Umwelttechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften:		

	Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Produktentwicklung und Produktion: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht General Engineering Science (7 Semester): Vertiefung Schiffbau: Pflicht General Engineering Science (7 Semester): Vertiefung Verfahrenstechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht Green Technologies: Energie, Wasser, Klima: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Logistik und Mobilität: Kernqualifikation: Pflicht Maschinenbau: Kernqualifikation: Pflicht Mechatronik: Kernqualifikation: Pflicht Orientierungsstudium: Kernqualifikation: Wahlpflicht Orientierungsstudium: Kernqualifikation: Wahlpflicht Schiffbau: Kernqualifikation: Pflicht Technomathematik: Kernqualifikation: Pflicht Verfahrenstechnik: Kernqualifikation: Pflicht Wirtschaftsingenieurwesen - Fachrichtung Logistik und Mobilität: Kernqualifikation: Pflicht
--	---

Lehrveranstaltung L0882: Betriebswirtschaftliche Übung	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Christoph Ihl, Katharina Roedelius
Sprachen	DE
Zeitraum	WiSe/SoSe
Inhalt	<p>In der betriebswirtschaftlichen Horsaalübung werden die Inhalte der Vorlesung durch praktische Beispiele und die Anwendung der diskutierten Werkzeuge vertieft.</p> <p>Bei angemessener Nachfrage wird parallel auch eine Problemorientierte Lehrveranstaltung angeboten, die Studierende alternativ wählen können. Hier bearbeiten die Studierenden in Gruppen ein selbstgewähltes Projekt, das sich thematisch mit der Ausarbeitung einer innovativen Geschäftsidee aus Sicht eines etablierten Unternehmens oder Startups befasst. Auch hier sollen die betriebswirtschaftlichen Grundkenntnisse aus der Vorlesung zum praktischen Einsatz kommen. Die Gruppenarbeit erfolgt unter Anleitung eines Mentors.</p>
Literatur	Relevante Literatur aus der korrespondierenden Vorlesung.

Lehrveranstaltung L0880: Grundlagen der Betriebswirtschaftslehre	
Typ	Vorlesung
SWS	3
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 48, Präsenzstudium 42
Dozenten	Prof. Christoph Ihl, Prof. Thorsten Blecker, Prof. Christian Lütjhe, Prof. Christian Ringle, Prof. Kathrin Fischer, Prof. Cornelius Herstatt, Prof. Wolfgang Kersten, Prof. Matthias Meyer, Prof. Thomas Wrona
Sprachen	DE
Zeitraum	WiSe/SoSe
Inhalt	<ul style="list-style-type: none"> • Die Abgrenzung der BWL von der VWL und die Gliederungsmöglichkeiten der BWL • Wichtige Definitionen aus dem Bereich Management und Wirtschaft • Die wichtigsten Unternehmensziele und ihre Einordnung sowie (Kern-) Funktionen der Unternehmung • Die Bereiche Produktion und Beschaffungsmanagement, der Begriff des Supply Chain Management und die Bestandteile einer Supply Chain • Die Definition des Begriffs Information, die Organisation des Informations- und Kommunikations (IuK)-Systems und Aspekte der Datensicherheit; Unternehmensstrategie und strategische Informationssysteme • Der Begriff und die Bedeutung von Innovationen, insbesondere Innovationschancen, -risiken und prozesse • Die Bedeutung des Marketing, seine Aufgaben, die Abgrenzung von B2B- und B2C-Marketing • Aspekte der Marketingforschung (Marktportfolio, Szenario-Technik) sowie Aspekte der strategischen und der operativen Planung und Aspekte der Preispolitik • Die grundlegenden Organisationsstrukturen in Unternehmen und einige Organisationsformen • Grundzüge des Personalmanagements • Die Bedeutung der Planung in Unternehmen und die wesentlichen Schritte eines Planungsprozesses • Die wesentlichen Bestandteile einer Entscheidungssituation sowie Methoden für Entscheidungsprobleme unter mehrfacher Zielsetzung, unter Ungewissheit sowie unter Risiko • Grundlegende Methoden der Finanzmathematik • Die Grundlagen der Buchhaltung, der Bilanzierung und der Kostenrechnung • Die Bedeutung des Controlling im Unternehmen und ausgewählte Methoden des Controlling • Die wesentlichen Aspekte von Entrepreneurship-Projekten <p>Neben der Vorlesung, die die Fachinhalte vermittelt, erarbeiten die Studierenden selbstständig in Gruppen einen Business-Plan für ein Gründungsprojekt. Dafür wird auch das wissenschaftliche Arbeiten und Schreiben gezielt unterstützt.</p>
Literatur	<p>Bamberg, G., Coenenberg, A.: Betriebswirtschaftliche Entscheidungslehre, 14. Aufl., München 2008</p> <p>Eisenführ, F., Weber, M.: Rationales Entscheiden, 4. Aufl., Berlin et al. 2003</p> <p>Heinhold, M.: Buchführung in Fallbeispielen, 10. Aufl., Stuttgart 2006.</p> <p>Kruschwitz, L.: Finanzmathematik. 3. Auflage, München 2001.</p> <p>Pellens, B., Fülbier, R. U., Gassen, J., Sellhorn, T.: Internationale Rechnungslegung, 7. Aufl., Stuttgart 2008.</p> <p>Schweitzer, M.: Planung und Steuerung, in: Bea/Friedl/Schweitzer: Allgemeine Betriebswirtschaftslehre, Bd. 2: Führung, 9. Aufl., Stuttgart 2005.</p> <p>Weber, J., Schäffer, U. : Einführung in das Controlling, 12. Auflage, Stuttgart 2008.</p> <p>Weber, J./Weißenberger, B.: Einführung in das Rechnungswesen, 7. Auflage, Stuttgart 2006.</p>

Modul M1432: Programmierparadigmen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Programmierparadigmen (L2169)	Vorlesung	2	2
Programmierparadigmen (L2170)	Hörsaalübung	1	1
Programmierparadigmen (L2171)	Laborpraktikum	2	3
Modulverantwortlicher	NN		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Veranstaltung Prozedurale Programmierung oder gleichwertige Programmierkenntnisse in imperativer Programmierung		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Die Studenten haben ein grundlegendes Verständnis über die objektorientierte und die generische Programmierung erworben und können diese in eigenen Programmierprojekten umsetzen. Sie können eigene Klassenhierarchien erstellen und verschiedene Formen der Vererbung unterscheiden. Sie haben ein grundlegendes Verständnis des Polymorphismus und können zwischen Laufzeit- und Compilerzeit-Polymorphismus unterscheiden. Die Studenten sind mit dem Konzept der Datenkapselung vertraut und können Schnittstellen in private und öffentliche Methoden unterteilen. Sie können mit Exceptions umgehen und nutzen generische Programmierung um Datenstrukturen zu verallgemeinern. Die Studenten können die Vor- und Nachteile der beiden Programmierparadigmen</p> <p><i>Fertigkeiten</i> Die Studenten können eine mittelgroße Problemstellung in Teilprobleme zerlegen und darauf aufbauend eigene Klassen in einer objektorientierten Programmiersprache erstellen. Sie können dabei ein öffentliche und private Schnittstellen entwerfen und die Implementierung durch Abstraktion generisch und erweiterbar umsetzen. Sie können verschiedene Sprachkonstrukte einer modernen Programmiersprache unterscheiden und diese geeignet in der Implementierung nutzen. Sie können Unit Tests entwerfen und implementieren.</p> <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i> Studierende können in Teams arbeiten und in Foren kommunizieren.</p> <p><i>Selbstständigkeit</i> In Programmierpraktikum lernen die Studenten unter Aufsicht die objektorientierte Programmierung. In Übungen entwickeln sie individuell und unabhängig Lösungen und erhalten Feedback.</p>		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Technomathematik: Kernqualifikation: Pflicht		

Lehrveranstaltung L2169: Programmierparadigmen	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Grundlegende Idee der Objektorientierten Programmierung • Klassen und Objekte • Vererbung (einfach, mehrfach) • Schnittstellen (Interfaces) • Datenkapselung (private / public usw.) • Ausnahmebehandlung (Exceptions) • Generische Programmierung und deren Umsetzung im Compiler • Exkurs in die Programmierung mit dynamisch getypten Programmiersprachen
Literatur	Skript

Lehrveranstaltung L2170: Programmierparadigmen	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Grundlegende Idee der Objektorientierten Programmierung • Klassen und Objekte • Vererbung (einfach, mehrfach) • Schnittstellen (Interfaces) • Datenkapselung (private / public usw.) • Ausnahmebehandlung (Exceptions) • Generische Programmierung und deren Umsetzung im Compiler • Exkurs in die Programmierung mit dynamisch getypten Programmiersprachen
Literatur	Skript

Lehrveranstaltung L2171: Programmierparadigmen	
Typ	Laborpraktikum
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Dozenten des SD E
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Grundlegende Idee der Objektorientierten Programmierung • Klassen und Objekte • Vererbung (einfach, mehrfach) • Schnittstellen (Interfaces) • Datenkapselung (private / public usw.) • Ausnahmebehandlung (Exceptions) • Generische Programmierung und deren Umsetzung im Compiler • Exkurs in die Programmierung mit dynamisch getypten Programmiersprachen
Literatur	Skript

Modul M1732: Mathematics III (EN)			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Analysis III (EN) (L2790)	Vorlesung	2	2
Analysis III (EN) (L2791)	Hörsaalübung	1	1
Analysis III (EN) (L2792)	Gruppenübung	1	1
Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (EN) (L2793)	Vorlesung	2	2
Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (EN) (L2794)	Hörsaalübung	1	1
Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (EN) (L2795)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Anusch Taraz		
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 128, Präsenzstudium 112		
Leistungspunkte	8		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Engineering Science: Kernqualifikation: Pflicht		

Lehrveranstaltung L2790: Analysis III (English)	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L2791: Analysis III (English)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2792: Analysis III (English)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2793: Differential Equations 1 (Ordinary Differential Equations)	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	WiSe
Inhalt	
Literatur	

Lehrveranstaltung L2794: Differential Equations 1 (Ordinary Differential Equations)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2795: Differential Equations 1 (Ordinary Differential Equations)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0834: Computernetworks and Internet Security			
Lehrveranstaltungen			
Titel		Typ	SWS
Rechnernetze und Internet-Sicherheit (L1098)		Vorlesung	3
Rechnernetze und Internet-Sicherheit (L1099)		Gruppenübung	1
Modulverantwortlicher	Prof. Andreas Timm-Giel		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Basics of Computer Science		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Students are able to explain important and common Internet protocols in detail and classify them, in order to be able to analyse and develop networked systems in further studies and job.		
<i>Fertigkeiten</i>	Students are able to analyse common Internet protocols and evaluate the use of them in different domains.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>			
<i>Selbstständigkeit</i>	Students can select relevant parts out of high amount of professional knowledge and can independently learn and understand it.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Wahlpflicht Elektrotechnik: Kernqualifikation: Wahlpflicht Engineering Science: Vertiefung Mechatronics: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L1098: Computer Networks and Internet Security	
Typ	Vorlesung
SWS	3
LP	5
Arbeitsaufwand in Stunden	Eigenstudium 108, Präsenzstudium 42
Dozenten	Prof. Andreas Timm-Giel, Prof. Dieter Gollmann, Dr.-Ing. Koojana Kuladinithi
Sprachen	EN
Zeitraum	WiSe
Inhalt	<p>In this class an introduction to computer networks with focus on the Internet and its security is given. Basic functionality of complex protocols are introduced. Students learn to understand these and identify common principles. In the exercises these basic principles and an introduction to performance modelling are addressed using computing tasks and (virtual) labs.</p> <p>In the second part of the lecture an introduction to Internet security is given.</p> <p>This class comprises:</p> <ul style="list-style-type: none"> • Application layer protocols (HTTP, FTP, DNS) • Transport layer protocols (TCP, UDP) • Network Layer (Internet Protocol, routing in the Internet) • Data link layer with media access at the example of Ethernet • Multimedia applications in the Internet • Network management • Internet security: IPSec • Internet security: Firewalls
Literatur	<ul style="list-style-type: none"> • Kurose, Ross, Computer Networking - A Top-Down Approach, 6th Edition, Addison-Wesley • Kurose, Ross, Computernetzwerke - Der Top-Down-Ansatz, Pearson Studium; Auflage: 6. Auflage • W. Stallings: Cryptography and Network Security: Principles and Practice, 6th edition <p>Further literature is announced at the beginning of the lecture.</p>

Lehrveranstaltung L1099: Computer Networks and Internet Security	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Andreas Timm-Giel, Prof. Dieter Gollmann
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0730: Technische Informatik			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Technische Informatik (L0321)		Vorlesung	3 4
Technische Informatik (L0324)		Gruppenübung	1 2
Modulverantwortlicher	Prof. Heiko Falk		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Grundkenntnisse der Elektrotechnik		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Dieses Modul vermittelt Grundkenntnisse der Funktionsweise von Rechensystemen. Abgedeckt werden die Ebenen von der Assemblerprogrammierung bis zur Gatterebene. Das Modul behandelt folgende Inhalte:		
<i>Wissen</i>	<ul style="list-style-type: none"> • Einführung • Kombinatorische Logik: Gatter, Boolesche Algebra, Schaltfunktionen, Synthese von Schaltungen, Schaltnetze • Sequentielle Logik: Flip-Flops, Schaltwerke, systematischer Schaltwerkwurf • Technologische Grundlagen • Rechnerarithmetik: Ganzzahlige Addition, Subtraktion, Multiplikation und Division • Grundlagen der Rechnerarchitektur: Programmiermodelle, MIPS-Einzelzyklusmaschine, Pipelining • Speicher-Hardware: Speicherhierarchien, SRAM, DRAM, Caches • Ein-/Ausgabe: I/O aus Sicht der CPU, Prinzipien der Datenübergabe, Point-to-Point Verbindungen, Busse 		
<i>Fertigkeiten</i>	<p>Die Studierenden fassen ein Rechensystem aus der Perspektive des Architekten auf, d.h. sie erkennen die interne Struktur und den physischen Aufbau von Rechensystemen. Die Studierenden können analysieren, wie hochspezifische und individuelle Rechner aus einer Sammlung gängiger Einzelkomponenten zusammengesetzt werden. Sie sind in der Lage, die unterschiedlichen Abstraktionsebenen heutiger Rechensysteme - von Gattern und Schaltungen bis hin zu Prozessoren - zu unterscheiden und zu erklären.</p> <p>Nach erfolgreichem Besuch der Veranstaltung sind die Studierenden in der Lage, die Wechselwirkungen zwischen einem physischen Rechensystem und der darauf ausgeführten Software beurteilen zu können. Insbesondere sollen sie die Konsequenzen der Ausführung von Software in den hardwarenahen Schichten von der Assemblersprache bis zu Gattern erkennen können. Sie sollen so in die Lage versetzt werden, Auswirkungen unterer Schichten auf die Leistung des Gesamtsystems abzuschätzen und geeignete Optionen vorzuschlagen.</p>		
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	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Ja 10 %	Übungsaufgaben	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten, Inhalte der Vorlesung und Übungen		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Bauingenieurwesen: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Verfahrenstechnik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Produktentwicklung und Produktion: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Schiffbau: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Medizingenieurwesen: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Energie- und Umwelttechnik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Bioverfahrenstechnik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Elektrotechnik: Pflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Wahlpflicht Elektrotechnik: Kernqualifikation: Pflicht General Engineering Science (7 Semester): Vertiefung Bauingenieurwesen: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht		

General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht
General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Produktentwicklung und Produktion: Pflicht
General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Mechatronik: Kernqualifikation: Pflicht
Technomathematik: Vertiefung II. Informatik: Wahlpflicht

Lehrveranstaltung L0321: Technische Informatik	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Einführung • Kombinatorische Logik • Sequentielle Logik • Technologische Grundlagen • Zahlendarstellungen und Rechnerarithmetik • Grundlagen der Rechnerarchitektur • Speicher-Hardware • Ein-/Ausgabe
Literatur	<ul style="list-style-type: none"> • A. Clements. The Principles of Computer Hardware. 3. Auflage, Oxford University Press, 2000. • A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001. • D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.

Lehrveranstaltung L0324: Technische Informatik	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1423: Algorithmen und Datenstrukturen			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Algorithmen und Datenstrukturen (L2046)		Vorlesung	4 4
Algorithmen und Datenstrukturen (L2047)		Gruppenübung	1 2
Modulverantwortlicher	Prof. Matthias Mnich		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Diskrete Algebraische Strukturen • Mathematik I • Mathematik II • Prozedurale Programming • Objectorientierte Programmierung 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<ul style="list-style-type: none"> • Studierende können die grundlegenden Begriffe des Algorithmenentwurfs, der Algorithmenanalyse und Problemreduktionen 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. 		
<i>Wissen</i>			
<i>Fertigkeiten</i>	<ul style="list-style-type: none"> • Studierende können diskrete Entscheidungsprobleme, Such- und Optimierungsprobleme mit Hilfe der kennengelernten Konzepte 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. 		
Personale Kompetenzen	<ul style="list-style-type: none"> • Studierende sind in der Lage, in Teams 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. 		
<i>Sozialkompetenz</i>			
<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> • 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	Klausur		
Prüfungsdauer und -umfang	60 min		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht Wirtschaftsingenieurwesen - Fachrichtung Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht		

Lehrveranstaltung L2046: Algorithmen und Datenstrukturen	
Typ	Vorlesung
SWS	4
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 64, Präsenzstudium 56
Dozenten	Prof. Matthias Mnich
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Sortieren durch Einfügen • Registermaschinen • Asymptotische Analyse, Landau Notation • Polynomialzeit Algorithmen and NP-Vollständigkeit • Divide-and-conquer, Merge sort • Strassens Algorithmus • Greedy Algorithmen • Dynamische Programmierung • Quicksort • AVL-trees, B-trees • Hashing • Tiefensuche und Breitensuche • Kürzeste Wege • Fluss Probleme, Ford-Fulkerson Algorithmus
Literatur	<ul style="list-style-type: none"> • T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms. MIT Press, 2013 • S. Skiena: The Algorithm Design Manual. Springer, 2008 • J. M. Kleinberg and É. Tardos. Algorithm Design. Addison-Wesley, 2005.

Lehrveranstaltung L2047: Algorithmen und Datenstrukturen	
Typ	Gruppenübung
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 M0727: Stochastik				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Stochastik (L0777)		Vorlesung	2	4
Stochastik (L0778)		Gruppenübung	2	2
Modulverantwortlicher	Prof. Matthias Schulte			
Zulassungsvoraussetzungen	Keine			
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Analysis • Aussagenlogik • Diskrete Algebraische Strukturen (Kombinatorik) 			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz	<ul style="list-style-type: none"> • Studierende können die grundlegenden Begriffe der Statistik benennen und anhand von Beispielen erklären. • Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern. • Studierende kennen Beweisstrategien und können diese wiedergeben. 			
<i>Wissen</i>				
<i>Fertigkeiten</i>				
Personale Kompetenzen				
<i>Sozialkompetenz</i>	<ul style="list-style-type: none"> • Studierende können in heterogen zusammengesetzten Teams (z.B. an Hausaufgaben) zusammenarbeiten und ihre Ergebnisse vor der Gruppe präsentieren. • Sie können sich dabei insbesondere gegenseitig neue Konzepte erklären und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen. 			
<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> • Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen. • Studierende können ihr Wissen mit den Inhalten anderer Veranstaltungen in Verbindung bringen. • 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	Klausur			
Prüfungsdauer und -umfang	120 min			
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht Theoretischer Maschinenbau: Kernqualifikation: Wahlpflicht Wirtschaftsingenieurwesen - Fachrichtung Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht			

Lehrveranstaltung L0777: Stochastik	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Prof. Matthias Schulte
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Wahrscheinlichkeitsdefinitionen, bedingte Wahrscheinlichkeiten • Zufallsvariablen • Unabhängigkeit • Verteilungs- und Dichtefunktionen • Kenngrößen: Erwartungswert, Varianz, Standardabweichung, Momente • Multivariate Verteilungen • Gesetz der großen Zahlen und zentraler Grenzwertsatz • Grundbegriffe zu stochastischen Prozessen • Grundkonzepte der Statistik (Punktschätzer, Konfidenzintervalle und Hypothesentests)
Literatur	<ol style="list-style-type: none"> 1. Methoden der statistischen Inferenz, Likelihood und Bayes, Held, L., Spektrum 2008 2. Stochastik für Informatiker, Dümbgen, L., Springer 2003 3. Statistik: Der Weg zur Datenanalyse, Fahrmeir, L., Künstler R., Pigeot, I, Tutz, G., Springer 2010 4. Stochastik, Georgii, H.-O., deGruyter, 2009 5. Probability and Random Processes, Grimmett, G., Stirzaker, D., Oxford University Press, 2001 6. Programmieren mit R, Ligges, U., Springer 2008

Lehrveranstaltung L0778: Stochastik	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Matthias Schulte
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0732: Software Engineering			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Software-Engineering (L0627)		Vorlesung	2 3
Software-Engineering (L0628)		Gruppenübung	2 3
Modulverantwortlicher	Prof. Sibylle Schupp		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Automata theory and formal languages • Procedural programming or Functional programming • Object-oriented programming, algorithms, and data structures 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Students explain the phases of the software life cycle, describe the fundamental terminology and concepts of software engineering, and paraphrase the principles of structured software development. They give examples of software-engineering tasks of existing large-scale systems. They write test cases for different test strategies and devise specifications or models using different notations, and critique both. They explain simple design patterns and the major activities in requirements analysis, maintenance, and project planning.</p> <p><i>Fertigkeiten</i> For a given task in the software life cycle, students identify the corresponding phase and select an appropriate method. They choose the proper approach for quality assurance. They design tests for realistic systems, assess the quality of the tests, and find errors at different levels. They apply and modify non-executable artifacts. They integrate components based on interface specifications.</p> <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i> Students practice peer programming. They explain problems and solutions to their peer. They communicate in English.</p> <p><i>Selbstständigkeit</i> Using on-line quizzes and accompanying material for self study, students can assess their level of knowledge continuously and adjust it appropriately. Working on exercise problems, they receive additional feedback.</p>		
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	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Kernqualifikation: Pflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0627: Software Engineering	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Software Life Cycle Models (Waterfall, V-Model, Evolutionary Models, Incremental Models, Iterative Models, Agile Processes) • Requirements (Elicitation Techniques, UML Use Case Diagrams, Functional and Non-Functional Requirements) • Specification (Finite State Machines, Extended FSMs, Petri Nets, Behavioral UML Diagrams, Data Modeling) • Design (Design Concepts, Modules, (Agile) Design Principles) • Object-Oriented Analysis and Design (Object Identification, UML Interaction Diagrams, UML Class Diagrams, Architectural Patterns) • Testing (Blackbox Testing, Whitebox Testing, Control-Flow Testing, Data-Flow Testing, Testing in the Large) • Maintenance and Evolution (Regression Testing, Reverse Engineering, Reengineering) • Project Management (Blackbox Estimation Techniques, Whitebox Estimation Techniques, Project Plans, Gantt Charts, PERT Charts)
Literatur	Kassem A. Saleh, Software Engineering, J. Ross Publishing 2009.

Lehrveranstaltung L0628: Software Engineering	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0852: Graphentheorie und Optimierung			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Graphentheorie und Optimierung (L1046)		Vorlesung	2 3
Graphentheorie und Optimierung (L1047)		Gruppenübung	2 3
Modulverantwortlicher	Prof. Anusch Taraz		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Diskrete Algebraische Strukturen • Mathematik I 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<ul style="list-style-type: none"> • Studierende können die grundlegenden Begriffe der Graphentheorie und 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. 		
<i>Wissen</i>			
<i>Fertigkeiten</i>			
Personale Kompetenzen			
<i>Sozialkompetenz</i>	<ul style="list-style-type: none"> • Studierende können Aufgabenstellungen der Graphentheorie und 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. 		
<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> • 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. 		
<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> • 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	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Pflicht Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht Logistik und Mobilität: Vertiefung Verkehrsplanung und -systeme: Wahlpflicht Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht Wirtschaftsingenieurwesen - Fachrichtung Logistik und Mobilität: Vertiefung Verkehrsplanung und -systeme: Wahlpflicht Wirtschaftsingenieurwesen - Fachrichtung Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht		

Lehrveranstaltung L1046: Graphentheorie und Optimierung	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Anusch Taraz
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Graphen, Durchlaufen von Graphen, Bäume • Planare Graphen • Kürzeste Wege • Minimale Spannbäume • Maximale Flüsse und minimale Schnitte • Sätze von Menger, König-Egervary, Hall • NP-vollständige Probleme • Backtracking und Heuristiken • Lineare Programmierung • Dualität • Ganzzahlige lineare Programmierung
Literatur	<ul style="list-style-type: none"> • M. Aigner: Diskrete Mathematik, Vieweg, 2004 • T. Cormen, Ch. Leiserson, R. Rivest, C. Stein: Algorithmen - Eine Einführung, Oldenbourg, 2013 • J. Matousek und J. Nešetřil: Diskrete Mathematik, Springer, 2007 • A. Steger: Diskrete Strukturen (Band 1), Springer, 2001 • A. Taraz: Diskrete Mathematik, Birkhäuser, 2012 • V. Turau: Algorithmische Graphentheorie, Oldenbourg, 2009 • K.-H. Zimmermann: Diskrete Mathematik, BoD, 2006

Lehrveranstaltung L1047: Graphentheorie und Optimierung	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Anusch Taraz
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0562: Berechenbarkeit und Komplexität			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Berechenbarkeit und Komplexität (L0166)		Vorlesung	2 3
Berechenbarkeit und Komplexität (L0167)		Gruppenübung	2 3
Modulverantwortlicher	Prof. Karl-Heinz Zimmermann		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Diskrete Algebraische Strukturen sowie Automatentheorie, Logik und Formale Sprachen.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Wissen: Die Studierenden kennen</p> <ul style="list-style-type: none"> • maschinennahe Modelle der Berechenbarkeit; • abstrakte funktionale Modelle der Berechenbarkeit; • das Konzept der universellen Berechenbarkeit und seine Beschreibung durch partiell-rekursive Funktionen; • das Konzept der Gödelisierung von Berechnungen sowie die Sätze von Kleene, Rice und Rice-Shapiro; • die Konzepte der entscheidbaren und semientscheidbaren Probleme; • die Wortprobleme in Semi-Thue-Systemen, Thue-Systemen, Halbgruppen und Post-Korrespondenz-Systemen; • Hilberts zehntes Problem; • die Komplexitätsklassen P und NP und deren Unterscheidung; • das Konzept der NP-Vollständigkeit sowie den Satz von Cook. <p><i>Fertigkeiten</i> Fertigkeiten: Die Studierenden können</p> <ul style="list-style-type: none"> • maschinennahe und abstrakte Modelle der Berechenbarkeit beschreiben; • Beziehungen zwischen den einzelnen Berechenbarkeitsbegriffen herstellen; • die grundlegenden Sätze von Kleene und Rice rekapitulieren und beweisen; • das Konzept der universellen Berechenbarkeit darlegen; • entscheidbare und semientscheidbare Probleme identifizieren und deren Bezug zu ähnlichen Problemen durch Reduktion herstellen; • die Komplexitätsklassen P und NP beschreiben; • NP-vollständige Probleme lokalisieren. <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.</p> <p><i>Selbstständigkeit</i> Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachbüchern und anderweitiger Literatur selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.</p>		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	60 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Kernqualifikation: Pflicht Data Science: Kernqualifikation: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0166: Berechenbarkeit und Komplexität	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Karl-Heinz Zimmermann
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L0167: Berechenbarkeit und Komplexität	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Karl-Heinz Zimmermann
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	
Literatur	

Modul M0873: Software-Fachpraktikum			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Modulverantwortlicher	Prof. Karl-Heinz Zimmermann		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Grundlagen der Softwaretechnik		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Die Studierenden kennen wichtige Aspekte und typische Phasen der Software-Projektentwicklung (wie Planung, Spezifikation, Implementierung, Validierung, Wartung und Dokumentation), die in der Praxis mehr oder weniger ausgeprägt zum Tragen kommen.		
<i>Fertigkeiten</i>	Die Studierenden können typische Stufen der Software-Projektentwicklung beschreiben und eng umrissene Teilaufgaben in einem Software-Projekt innerhalb einer Gruppe oder eigenständig bearbeiten.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben aus der Software-Entwicklung alleine oder in einer Gruppe zu bearbeiten und die Resultate geeignet zu präsentieren.		
<i>Selbstständigkeit</i>	Die Studierenden sind nach Abschluss des Moduls imstande, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbstständig zu erarbeiten, das erworbene Wissen zusammenzufassen und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.		
Arbeitsaufwand in Stunden	Eigenstudium 180, Präsenzstudium 0		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Schriftliche Ausarbeitung (laut FPrO)		
Prüfungsdauer und -umfang	Die Ausarbeitung wird von der Betreuerin bzw. dem Betreuer der Bachelorarbeit bewertet.		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Pflicht		

Modul M1578: Seminare Informatik			
Lehrveranstaltungen			
Titel		Typ	SWS
Seminar Informatik I (L2362)		Seminar	2
Seminar Informatik II (L2361)		Seminar	2
LP			
			3
Modulverantwortlicher	Prof. Karl-Heinz Zimmermann		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Grundlegende Module aus der Informatik und Mathematik auf Bachelorebene.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Die Studierenden können		
	<ul style="list-style-type: none"> • ein spezifisches Thema der Informatik erklären, • komplexe Sachverhalte beschreiben, • unterschiedliche Standpunkte darlegen und kritisch bewerten. 		
<i>Fertigkeiten</i>	Die Studierenden können		
	<ul style="list-style-type: none"> • 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,		
	<ul style="list-style-type: none"> • 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,		
	<ul style="list-style-type: none"> • 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	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Referat		
Prüfungsdauer und -umfang	x		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Kernqualifikation: Pflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht		

Lehrveranstaltung L2362: Seminar Informatik I	
Typ	Seminar
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Karl-Heinz Zimmermann
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2361: Seminar Informatik II	
Typ	Seminar
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Karl-Heinz Zimmermann
Sprachen	DE/EN
Zeitraum	WiSe/SoSe
Inhalt	
Literatur	

Modul M1620: Ethik in der Informationstechnologie			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Ethik in der Informationstechnologie (L2450)	Vorlesung	2	3
Ethik in der Informationstechnologie (L2451)	Seminar	2	3
Modulverantwortlicher	NN		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Keine		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Die Studierenden kennen		
<i>Wissen</i>	<ul style="list-style-type: none"> • Ethische Grundpositionen • Bedeutungen des Informationsbegriffs und ihren historischen Wandel • Ethische Grundprobleme der Informationstechnologie (Entscheidungsautonomie von Algorithmen und künstlicher Intelligenz; Macht durch Zugang und Nutzen von Daten etc.) • Auswirkungen einer zunehmenden Erhebung und Analyse von Daten auf Individuen und moderne Gesellschaften • Datenschutzrichtlinien im Allgemeinen und in spezifischen Anwendungsgebieten (Beispiel: medizinische Daten) • Auswirkungen von Fehlern in Softwaresystemen • Die ethischen Leitlinien der Deutschen Gesellschaft für Informatik und die Empfehlungen zur Guten wissenschaftlichen Praxis der DFG 		
<i>Fertigkeiten</i>	Die Studierenden können		
	<ul style="list-style-type: none"> • ethische Grundpositionen in der Analyse von Beispielen aus der Geschichte und Gegenwart der Informatik und Data Science anwenden. • ethische Konflikte bezüglich dem Sammeln und der Verarbeitung von Daten erkennen und beschreiben • ihr eigenes Handeln bei der Erfassung, Verarbeitung und Analyse von Daten und die dessen Folgen reflektieren • Datenschutzrichtlinien berücksichtigen und die Konformität von Softwaresystemen mit Datenschutzrichtlinien bewerten. • die Auswirkung von Softwarefehlern in einem konkreten Anwendungsgebiet einschätzen und geeignete Maßnahmen zur Minimierung von Fehlern umsetzen 		
Personale Kompetenzen	Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben alleine oder in der Gruppe zu bearbeiten und geeignet zu präsentieren.		
<i>Sozialkompetenz</i>			
<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	Fachtheoretisch-fachpraktische Arbeit		
Prüfungsdauer und -umfang	-		
Zuordnung zu folgenden Curricula	Computer Science: Kernqualifikation: Wahlpflicht Data Science: Kernqualifikation: Pflicht		

Lehrveranstaltung L2450: Ethik in der Informationstechnologie	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	NN
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	<p>Vorlesung:</p> <ul style="list-style-type: none"> • Einführung in ethische Grundpositionen an Fallbeispielen aus der Informationstechnologie • Was sind Informationen? Definitionen des Informationsbegriffs und ihr historisches Wandel • Besondere Probleme der Ethik der Informationstechnologie: Entscheidungsautonomie von KI; Verantwortung für und Auswirkungen von Software-Fehler etc. • Einführung in grundlegende Positionen der Techniksoziologie an Fallbeispielen aus der Informationstechnologie (Datafizierung; Digitalisiertes Selbst etc.) • Datenschutzrichtlinien und Anwendungsbeispiele • Inhalte und Probleme ethischer Leitlinien für Informatikerinnen und Informatiker (Deutsche Gesellschaft für Informatik, DFG etc.) <p>Seminar:</p> <ul style="list-style-type: none"> • Vertiefung ausgewählter Aspekte der Vorlesung • Diskussion aktueller Entwicklungen in der Ethik der Informationstechnologie
Literatur	Wird zu Beginn der Lehrveranstaltung bekannt gegeben.

Lehrveranstaltung L2451: Ethik in der Informationstechnologie	
Typ	Seminar
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	NN
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Fachmodule der Vertiefung I. Computer- und Software-Engineering

In dieser Vertiefung finden sich die Gebiete Softwaretechnik und Technische Informatik.

Die Softwaretechnik beschäftigt sich mit Planung, Entwurf, Analyse, Betrieb und Dokumentation von Softwaresystemen sowie der Spezifikation der dazu notwendigen Datenstrukturen. Die Vertiefung bietet einerseits Module aus dem Kernbereich der Softwaretechnik, wie Compilerbau, Datenbanken und Software-Entwicklung, und andererseits anwendungsorientierte Module, wie Verteilte Systeme, Kombinatorische Strukturen und Algorithmen sowie Rechnergestützte Geometrie.

Die Technische Informatik befasst sich mit Aufbau, Entwurf, Organisation, Bewertung und Betrieb von Rechnersystemen. Sie ist Schnittstelle zwischen Informatik und Elektrotechnik und beschreibt Rechnersysteme hauptsächlich aus elektrologischer Sicht. Kenntnisse aus den grundständigen Veranstaltungen über Elektrotechnik werden nicht vorausgesetzt. Die Vertiefung bietet zum einen Module aus dem Kernbereich der Technischen Informatik, wie Recherarchitektur, Eingebettete Systeme sowie ein entsprechendes Labor-Praktikum, und andererseits anwendungsorientierte Module, wie Medizintechnische Systeme und die klassische Nachrichtentechnik. Die Einführungsveranstaltung über Quantenmechanik spielt eine fundamentale Rolle beim Aufbau von Quantenrechnern.

Modul M0625: Databases			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Datenbanken (L0337)	Vorlesung	4	5
Datenbanken (L1150)	Projekt-/problembasierte Lehrveranstaltung	1	1
Modulverantwortlicher	Prof. Stefan Schulte		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Students should have basic knowledge in the following areas: <ul style="list-style-type: none"> • Discrete Algebraic Structures • Procedural Programming • Automata Theory and Formal Languages • Programming Paradigms 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> After successful completion of the course, students know:</p> <ul style="list-style-type: none"> • Design instruments for relational databases • The relational model • Relational query languages, especially SQL • Requirements on data integrity • Possibilities for query optimization • Aspects of transaction handling, fault handling and concurrency/synchronization in database systems • Specific attributes and differences of object-oriented and object-relational databases • Paradigms and concepts of current technologies for data modelling and database systems <p><i>Fertigkeiten</i> The students acquire the ability to model a database and to work with it. This comprises especially the application of design methodologies and query and definition languages. Furthermore, students are able to apply basic functionalities needed to run a database.</p>		
Personale Kompetenzen	<p><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.</p> <p><i>Selbstständigkeit</i> Students are able to independently investigate a complex problem and assess which competencies are required to solve it.</p>		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Data Science: Kernqualifikation: Pflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0337: Databases	
Typ	Vorlesung
SWS	4
LP	5
Arbeitsaufwand in Stunden	Eigenstudium 94, Präsenzstudium 56
Dozenten	Prof. Stefan Schulte
Sprachen	EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Architecture of database systems, conceptual data modeling with the Entity Relationship (ER) modeling language • Relational data model, referential integrity, keys, foreign keys, functional dependencies (FDs), canonical mapping of entity types and relationship into the relational data model, anomalies • Relational algebra as a simple query language • Dependency theory, FD closure, canonical cover of FD set, decomposition of relational schemata, multivalued dependencies, normalization, inclusion dependencies • Practical query languages and integrity constraints w/o considering a conceptual domain model: SQL • Storage structures, database implementation architecture • Index structures • Query processing • Query optimization • Transactions and recovery • Query languages with recursion and consideration of a simple conceptual domain model: Datalog • Semi-naive evaluation strategy, magic sets transformation • Information integration, declarative schema transformation (LAV, GAV), distributed database systems • Description logics, syntax, semantics, decision problems, decision algorithms for Abox satisfiability • Ontology based data access (OBDA), DL-Lite for formalizing ER diagrams • Complexity measure: Data complexity • Semistructured databases and query languages: XML and XQuery
Literatur	<ol style="list-style-type: none"> 1. A. Kemper, A. Eickler, Datenbanksysteme - n. Auflage, Oldenbourg, 2010 2. S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995 3. Database Systems, An Application Oriented Approach, Pearson International Edition, 2005 4. H. Garcia-Molina, J.D. Ullman, J. Widom, Database Systems: The Complete Book, Prentice Hall, 2002

Lehrveranstaltung L1150: Databases	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Stefan Schulte
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0971: Betriebssysteme			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Betriebssysteme (L1153)	Vorlesung	2	3
Betriebssysteme (L1154)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Volker Turau		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Prozedurales Programmieren • Objekt-orientierte Programmierung, Algorithmen und Datenstrukturen • Erfahrung in der Anwendung von betriebssystemnahen Werkzeugen wie Editoren, Linker, Compiler • Erfahrung im Umgang mit C-Bibliotheken 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Studierende können die wichtigsten Abstraktion von Betriebssystem erklären (Prozess, virtueller Speicher, Datei, Deadlock, Livelock). Sie sind in der Lage, die Prozesszustände und die dazugehörigen Übergänge zu beschreiben. Sie kennen die wichtigsten Architekturvarianten von Betriebssystemen und können existierende Betriebssysteme diesen Varianten zuordnen. Die Teilnehmer sind in der Lage, nebenläufige Programm mittels Threads, conditional Variablen und Semaphoren zu erstellen. Sie können mehrere Varianten zur Realisierung von Filesystemen erläutern. Des Weiteren können sie mindestens drei Scheduling Algorithmen erläutern.</p> <p><i>Fertigkeiten</i> Studierende können die POSIX Bibliotheken zur nebenläufigen Programmierung korrekt und effizient einsetzen. Sie sind in der Lage für eine Scheduling Aufgabe unter gegebenen Randbedingungen die Effizienz eines Scheduling-Algorithmus zu beurteilen.</p>		
Personale Kompetenzen	<p><i>Sozialkompetenz</i></p> <p><i>Selbstständigkeit</i></p>		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L1153: Betriebssysteme	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Volker Turau
Sprachen	DE
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Architekturen für Betriebssysteme • Prozesse • Nebenläufigkeit • Verklemmungen • Speicherverwaltung • Scheduling • Dateisysteme
Literatur	1. Operating Systems, William Stallings, Pearson International Edition 2. Moderne Betriebssysteme, Andrew Tanenbaum, Pearson Studium

Lehrveranstaltung L1154: Betriebssysteme	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Volker Turau
Sprachen	DE
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1586: Wissenschaftliche Programmierung			
Lehrveranstaltungen			
Titel		Typ	SWS
Wissenschaftliche Programmierung (L2405)		Vorlesung	3
Wissenschaftliche Programmierung (L2406)		Gruppenübung	2
Modulverantwortlicher	Prof. Tobias Knopp		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Prozedurale Programmierung, Lineare Algebra		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Die Studierenden		
<i>Wissen</i>	<ul style="list-style-type: none"> können wissenschaftliche Probleme in einer modernen Programmiersprache effizient lösen. sind mit dem Konzept der reproduzierbaren Wissenschaft vertraut. können mit mehrdimensionalen Arrays, sparse Arrays, Data Frames (tabellenförmige Daten) und Missing Data umgehen. Sie kennen die Vor- und Nachteile spezifischer Datenstrukturen. kennen verschiedene Möglichkeiten um Daten, Datenbeziehungen und Fehlermaße geeignet darzustellen. Sie kennen bekannte Datenformate zur Speicherung von wissenschaftlichen Daten und können für spezifische Daten ein geeignetes Format auswählen. 		
<i>Fertigkeiten</i>	<p>Sie sind in der Lage</p> <ul style="list-style-type: none"> komplexe Probleme aus einer mathematischen Formulierung in ein geeignetes Programm zu übersetzen. ein komplexes Problem in Teilprobleme aufzuteilen welche modular umgesetzt werden können. numerische Standardprobleme zu identifizieren und hierfür geeignete Standardalgorithmen nutzen, die in Bibliotheken vorhanden sind. wartbaren Programmcode zu schreiben, dessen Korrektheit durch geeignete Tests überprüft wird. die Laufzeit von Programmen zu messen, Flaschenhalse zu identifizieren und geeignete Beschleunigungstechniken anzuwenden. 		
Personale Kompetenzen	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>Sozialkompetenz</i>	Die Studierenden sind in der Lage ein komplexes Problem eigenständig zu untersuchen und einzuschätzen, welche Kompetenzen zur Lösung des Problems benötigt werden.		
<i>Selbstständigkeit</i>			
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Data Science: Kernqualifikation: Pflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L2405: Wissenschaftliche Programmierung	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Tobias Knopp
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> Elementare Datentypen und der Zusammenhang zur Mathematik Wissenschaftliche Datentypen: Mehrdimensionale Arrays, sparse Arrays, Data Frames, Missing Data Multiple Dispatch als effizientes Paradigma für die wissenschaftliche Programmierung Literate Programming Profiling und Benchmarks Beschleunigungstechniken: Caching, Multi-threading, SIMD, GPGPU Wissenschaftliche Datenformate: CSV, TOML, HDF5, und ausgewählte Beispiele Datenvisualisierung Numerische Standardtechniken und effiziente Programmbibliotheken (BLAS, LAPACK, FFTW, ...) Tests, Codeverwaltung, Dokumentation Reproduzierbare Wissenschaft
Literatur	Ben Lauwens, Allen Downey: Think Julia: How to Think Like a Computer Scientist

Lehrveranstaltung L2406: Wissenschaftliche Programmierung	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Tobias Knopp
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0791: Rechnerarchitektur			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Rechnerarchitektur (L0793)	Vorlesung	2	3
Rechnerarchitektur (L0794)	Projekt-/problembasierte Lehrveranstaltung	2	2
Rechnerarchitektur (L1864)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Heiko Falk		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Modul "Technische Informatik"		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	In diesem Modul werden fortgeschrittene Konzepte der Rechnerarchitektur vorgestellt. Am Anfang steht ein breiter Überblick über mögliche Programmiermodelle, wie sie für Universalrechner aber auch für spezielle Maschinen (z.B. Signalprozessoren) entwickelt wurden. Anschließend werden prinzipielle Aspekte der Mikroarchitektur von Prozessoren behandelt. Der Schwerpunkt liegt hierbei insbesondere auf dem sogenannten Pipelining und den in diesem Zusammenhang angewandten Methoden zur Beschleunigung der Befehlsausführung. Die Studierenden lernen Mechanismen zum dynamischen Scheduling, zur Sprungvorhersage, zu superskalaren Architekturen und zu Speicher-Hierarchien kennen.		
<i>Wissen</i>			
<i>Fertigkeiten</i>	Die Studierenden sind in der Lage, den Aufbau eines Prozessors zu erklären. Sie kennen die verschiedenen Architekturprinzipien und Programmiermodelle. Die Studierenden untersuchen verschiedene Strukturen von Pipeline-Architekturen und sind in der Lage, deren Konzepte zu erklären und im Hinblick auf Kriterien wie Performance und Energieeffizienz zu analysieren. Sie bewerten unterschiedliche Speicherarchitekturen, kennen parallele Rechnerarchitekturen und können zwischen Befehls- und Datenparallelität unterscheiden.		
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 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Nein 15 %	Fachtheoretisch- fachpraktische Studienleistung	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Min., Vorlesungsstoff + 4 Testate zur PBL "Rechnerarchitektur"		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Flugzeug-Systemtechnik: Kernqualifikation: Wahlpflicht Flugzeug-Systemtechnik: Vertiefung Avionik und eingebettete Systeme: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht		

Lehrveranstaltung L0793: Rechnerarchitektur	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Einführung • Grundlagen von VHDL • Programmiermodelle • Realisierung elementarer Datentypen • Dynamisches Scheduling • Sprungvorhersage • Superskalare Maschinen • Speicher-Hierarchien <p>Die Gruppenübungen vertiefen die Vorlesungsinhalte durch Bearbeiten und Besprechen von Übungsblättern und dienen somit zur Klausur-Vorbereitung. Der praktische Umgang mit Fragestellungen aus der Rechnerarchitektur wird in der FPGA-basierten PBL zur Rechnerarchitektur vermittelt, deren Teilnahme verpflichtend ist.</p>
Literatur	<ul style="list-style-type: none"> • D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005. • A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.

Lehrveranstaltung L0794: Rechnerarchitektur	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L1864: Rechnerarchitektur	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0953: Introduction to Information Security			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Einführung in die Informationssicherheit (L1114)		Vorlesung	2 3
Einführung in die Informationssicherheit (L1115)		Gruppenübung	2 3
Modulverantwortlicher	Prof. Riccardo Scandariato		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Basics of Computer Science		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Students can <ul style="list-style-type: none"> name the main security risks when using Information and Communication Systems and name the fundamental security mechanisms, describe commonly used methods for risk and security analysis, name the fundamental principles of data protection. 		
<i>Fertigkeiten</i>	Students can <ul style="list-style-type: none"> evaluate the strenghts and weaknesses of the fundamental security mechanisms and of the commonly used methods for risk and security analysis, apply the fundamental principles of data protection to concrete cases. 		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Students are capable of appreciating the impact of security problems on those affected and of the potential responsibilities for their resolution.		
<i>Selbstständigkeit</i>	None		
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 Data Science: Kernqualifikation: Pflicht		

Lehrveranstaltung L1114: Introduction to Information Security	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Riccardo Scandariato
Sprachen	EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> Fundamental concepts Passwords & biometrics Introduction to cryptography Sessions, SSL/TLS Certificates, electronic signatures Public key infrastructures Side-channel analysis Access control Privacy Software security basics Security management & risk analysis Security evaluation: Common Criteria
Literatur	D. Gollmann: Computer Security, Wiley & Sons, third edition, 2011 Ross Anderson: Security Engineering, Wiley & Sons, second edition, 2008

Lehrveranstaltung L1115: Introduction to Information 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 M0803: Embedded Systems			
Lehrveranstaltungen			
Titel		Typ	SWS
Eingebettete Systeme (L0805)		Vorlesung	3
Eingebettete Systeme (L0806)		Gruppenübung	1
Modulverantwortlicher	Prof. Heiko Falk		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	Computer Engineering		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i></p> <p>Embedded systems can be defined as information processing systems embedded into enclosing products. This course teaches the foundations of such systems. In particular, it deals with an introduction into these systems (notions, common characteristics) and their specification languages (models of computation, hierarchical automata, specification of distributed systems, task graphs, specification of real-time applications, translations between different models).</p> <p>Another part covers the hardware of embedded systems: Sensors, A/D and D/A converters, real-time capable communication hardware, embedded processors, memories, energy dissipation, reconfigurable logic and actuators. The course also features an introduction into real-time operating systems, middleware and real-time scheduling. Finally, the implementation of embedded systems using hardware/software co-design (hardware/software partitioning, high-level transformations of specifications, energy-efficient realizations, compilers for embedded processors) is covered.</p> <p><i>Fertigkeiten</i></p> <p>After having attended the course, students shall be able to realize simple embedded systems. The students shall realize which relevant parts of technological competences to use in order to obtain a functional embedded systems. In particular, they shall be able to compare different models of computations and feasible techniques for system-level design. They shall be able to judge in which areas of embedded system design specific risks exist.</p>		
Personale Kompetenzen	<p><i>Sozialkompetenz</i></p> <p>Students are able to solve similar problems alone or in a group and to present the results accordingly.</p> <p><i>Selbstständigkeit</i></p> <p>Students are able to acquire new knowledge from specific literature and to associate this knowledge with other classes.</p>		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Ja 10 %	Fachtheoretisch-fachpraktische Studienleistung	
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten, Inhalte der Vorlesung und Übungen		
Zuordnung zu folgenden Curricula	<p>Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht</p> <p>Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht</p> <p>Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht</p> <p>Elektrotechnik: Kernqualifikation: Wahlpflicht</p> <p>Engineering Science: Vertiefung Mechatronics: Wahlpflicht</p> <p>Flugzeug-Systemtechnik: Kernqualifikation: Wahlpflicht</p> <p>General Engineering Science (7 Semester): Vertiefung Mechatronics: Wahlpflicht</p> <p>Informatik-Ingenieurwesen: Kernqualifikation: Pflicht</p> <p>Mechatronics: Vertiefung Systementwurf: Wahlpflicht</p> <p>Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht</p> <p>Mechatronik: Kernqualifikation: Wahlpflicht</p> <p>Microelectronics and Microsystems: Vertiefung Embedded Systems: Wahlpflicht</p>		

Lehrveranstaltung L0805: Embedded Systems	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Heiko Falk
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Introduction • Specifications and Modeling • Embedded/Cyber-Physical Systems Hardware • System Software • Evaluation and Validation • Mapping of Applications to Execution Platforms • Optimization
Literatur	<ul style="list-style-type: none"> • Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2nd Edition, Springer, 2012., Springer, 2012.

Lehrveranstaltung L0806: Embedded Systems	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Heiko Falk
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0754: Compiler Construction			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Compilerbau (L0703)		Vorlesung	2 2
Compilerbau (L0704)		Gruppenübung	2 4
Modulverantwortlicher	Prof. Sibylle Schupp		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Practical programming experience • Automata theory and formal languages • Functional programming or procedural programming • Object-oriented programming, algorithms, and data structures • Basic knowledge of software engineering 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Students explain the workings of a compiler and break down a compilation task in different phases. They apply and modify the major algorithms for compiler construction and code improvement. They can re-write those algorithms in a programming language, run and test them. They choose appropriate internal languages and representations and justify their choice. They explain and modify implementations of existing compiler frameworks and experiment with frameworks and tools.		
<i>Fertigkeiten</i>	Students design and implement arbitrary compilation phases. They integrate their code in existing compiler frameworks. They organize their compiler code properly as a software project. They generalize algorithms for compiler construction to algorithms that analyze or synthesize software.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Students develop the software in a team. They explain problems and solutions to their team members. They present and defend their software in class. They communicate in English.		
<i>Selbstständigkeit</i>	Students develop their software independently and define milestones by themselves. They receive feedback throughout the entire project. They organize the software project so that they can assess their progress themselves.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Fachtheoretisch-fachpraktische Arbeit		
Prüfungsdauer und -umfang	Software (Compiler)		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Computer Science: Vertiefung I. Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0703: Compiler Construction	
Typ	Vorlesung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Lexical and syntactic analysis • Semantic analysis • High-level optimization • Intermediate languages and code generation • Compilation pipeline
Literatur	Alfred Aho, Jeffrey Ullman, Ravi Sethi, and Monica S. Lam, Compilers: Principles, Techniques, and Tools, 2nd edition Aarne Ranta, Implementing Programming Languages, An Introduction to Compilers and Interpreters, with an appendix coauthored by Markus Forsberg, College Publications, London, 2012

Lehrveranstaltung L0704: Compiler Construction	
Typ	Gruppenübung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1300: Software Development			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Softwareentwicklung (L1790)		Projekt-/problembasierte Lehrveranstaltung	2 5
Softwareentwicklung (L1789)		Vorlesung	1 1
Modulverantwortlicher	Prof. Sibylle Schupp		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Introduction to Software Engineering • Programming Skills • Experience with Developing Small to Medium-Size Programs 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i></p> <p>Students explain the fundamental concepts of agile methods, describe the process of test-driven development, and explain how continuous integration can be used in different scenarios. They give examples of selected pitfalls in software development, regarding scalability and other non-functional requirements. They write unit tests and build scripts and combine them in a corresponding integration environment. They explain major activities in requirements analysis, program comprehension, and agile project development.</p> <p><i>Fertigkeiten</i></p> <p>For a given task on a legacy system, students identify the corresponding parts in the system and select an appropriate method for understanding the details. They choose the proper approach of splitting a task in independent testable and extensible pieces and, thus, solve the task with proper methods for quality assurance. They design tests for legacy systems, create automated builds, and find errors at different levels. They integrate the resulting artifacts in a continuous development environment</p> <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i></p> <p>Students discuss different design decisions in a group. They defend their solutions orally. They communicate in English.</p> <p><i>Selbstständigkeit</i></p> <p>Using accompanying tools, students can assess their level of knowledge continuously and adjust it appropriately. Within limits, they can goals. Upon successful completion, students can identify and formulate concrete problems of software systems and propose solutions. Wit conduct independent studies to acquire the necessary competencies. They can devise plans to arrive at new solutions or assess existing ones</p>		
Arbeitsaufwand in Stunden	Eigenstudium 138, Präsenzstudium 42		
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 Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung I. Informatik: Wahlpflicht		

Lehrveranstaltung L1790: Software Development	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	2
LP	5
Arbeitsaufwand in Stunden	Eigenstudium 122, Präsenzstudium 28
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Agile Methods • Test-Driven Development and Unit Testing • Continuous Integration • Web Services • Scalability • From Defects to Failure
Literatur	<p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p>http://scrum-kompakt.de/</p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.</p>

Lehrveranstaltung L1789: Software Development	
Typ	Vorlesung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Sibylle Schupp
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Agile Methods • Test-Driven Development and Unit Testing • Continuous Integration • Web Services • Scalability • From Defects to Failure
Literatur	<p>Duvall, Paul M. Continuous Integration. Pearson Education India, 2007.</p> <p>Humble, Jez, and David Farley. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, 2010.</p> <p>Martin, Robert Cecil. Agile software development: principles, patterns, and practices. Prentice Hall PTR, 2003.</p> <p>http://scrum-kompakt.de/</p> <p>Myers, Glenford J., Corey Sandler, and Tom Badgett. The art of software testing. John Wiley & Sons, 2011.</p>

Fachmodule der Vertiefung II. Mathematik und Ingenieurwissenschaften

In dieser Vertiefung geht es vor allem um Mathematik unter Verwendung von Computern. Aufbauend auf den in den ersten vier Semestern vermittelten Grundlagen der höheren Mathematik werden weitergehende Veranstaltungen aus den Bereichen Analysis, Geometrie, Kombinatorik und Numerik angeboten.

Modul M0662: Numerical Mathematics I			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Numerische Mathematik I (L0417)	Vorlesung	2	3
Numerische Mathematik I (L0418)	Gruppenübung	2	3
Modulverantwortlicher	Prof. Sabine Le Borne		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Mathematik I + II for Engineering Students (german or english) o r Analysis & Linear Algebra I + II for Technomathematicians • basic MATLAB/Python knowledge 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Students are able to</p> <ul style="list-style-type: none"> • name numerical methods for interpolation, integration, least squares problems, eigenvalue problems, nonlinear root finding problems and to explain their core ideas, • repeat convergence statements for the numerical methods, • explain aspects for the practical execution of numerical methods with respect to computational and storage complexitx. <p><i>Fertigkeiten</i> Students are able to</p> <ul style="list-style-type: none"> • implement, apply and compare numerical methods using MATLAB/Python, • justify the convergence behaviour of numerical methods with respect to the problem and solution algorithm, • select and execute a suitable solution approach for a given problem. 		
Personale Kompetenzen	<p><i>Sozialkompetenz</i> Students are able to</p> <ul style="list-style-type: none"> • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms. <p><i>Selbstständigkeit</i> Students are capable</p> <ul style="list-style-type: none"> • to assess whether the supporting theoretical and practical excercises are better solved individually or in a team, • to assess their individual progress and, if necessary, to ask questions and seek help. 		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 Minuten		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Medizingenieurwesen: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Flugzeug-Systemtechnik: Wahlpflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Wahlpflicht Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Energietechnik: Wahlpflicht Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht Computer Science: Vertiefung Computermathematik: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Pflicht Elektrotechnik: Kernqualifikation: Wahlpflicht Engineering Science: Kernqualifikation: Pflicht Engineering Science: Kernqualifikation: Pflicht General Engineering Science (7 Semester): Kernqualifikation: Pflicht General Engineering Science (7 Semester): Vertiefung Informatik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht General Engineering Science (7 Semester): Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Wahlpflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht		

	Maschinenbau: Vertiefung Theoretischer Maschinenbau: Pflicht Maschinenbau: Vertiefung Energietechnik: Wahlpflicht Maschinenbau: Vertiefung Mechatronik: Wahlpflicht Theoretischer Maschinenbau: Technischer Ergänzungskurs Kernfächer: Wahlpflicht Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht
--	--

Lehrveranstaltung L0417: Numerical Mathematics I	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne
Sprachen	EN
Zeitraum	WiSe
Inhalt	<ol style="list-style-type: none"> 1. Finite precision arithmetic, error analysis, conditioning and stability 2. Linear systems of equations: LU and Cholesky factorization, condition 3. Interpolation: polynomial, spline and trigonometric interpolation 4. Nonlinear equations: fixed point iteration, root finding algorithms, Newton's method 5. Linear and nonlinear least squares problems: normal equations, Gram Schmidt and Householder orthogonalization, singular value decomposition, regularization, Gauss-Newton and Levenberg-Marquardt methods 6. Eigenvalue problems: power iteration, inverse iteration, QR algorithm 7. Numerical differentiation 8. Numerical integration: Newton-Cotes rules, error estimates, Gauss quadrature, adaptive quadrature
Literatur	<ul style="list-style-type: none"> • Gander/Gander/Kwok: Scientific Computing: An introduction using Maple and MATLAB, Springer (2014) • Stoer/Bulirsch: Numerische Mathematik 1, Springer • Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer

Lehrveranstaltung L0418: Numerical Mathematics I	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne, Dr. Jens-Peter Zemke
Sprachen	EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1730: Mathematics IV (EN)			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Differentialgleichungen 2 (Partielle Differentialgleichungen) (EN) (L2783)	Vorlesung	2	1
Differentialgleichungen 2 (Partielle Differentialgleichungen) (EN) (L2784)	Hörsaalübung	1	1
Differentialgleichungen 2 (Partielle Differentialgleichungen) (EN) (L2785)	Gruppenübung	1	1
Komplexe Funktionen (EN) (L2786)	Vorlesung	2	1
Komplexe Funktionen (EN) (L2787)	Hörsaalübung	1	1
Komplexe Funktionen (EN) (L2788)	Gruppenübung	1	1
Modulverantwortlicher	Prof. Anusch Taraz		
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 68, Präsenzstudium 112		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	120 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Wahlpflicht Engineering Science: Kernqualifikation: Pflicht Engineering Science: Vertiefung Elektrotechnik: Pflicht Engineering Science: Vertiefung Mechatronics: Wahlpflicht		

Lehrveranstaltung L2783: Differential Equations 2 (Partial Differential Equations) (English)	
Typ	Vorlesung
SWS	2
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 2, Präsenzstudium 28
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2784: Differential Equations 2 (Partial Differential Equations) (English)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2785: Differential Equations 2 (Partial Differential Equations) (English)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2786: Complex Functions (English)	
Typ	Vorlesung
SWS	2
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 2, Präsenzstudium 28
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	SoSe
Inhalt	
Literatur	

Lehrveranstaltung L2787: Complex Functions (English)	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L2788: Complex Functions (English)	
Typ	Gruppenübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Dozenten des Fachbereiches Mathematik der UHH
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0651: Rechnergestützte Geometrie			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Rechnergestützte Geometrie (L0393)	Vorlesung	2	4
Rechnergestützte Geometrie (L0394)	Gruppenübung	2	2
Modulverantwortlicher	Dr. Prashant Batra		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Lineare Algebra und Analytische Geometrie wie aus der schulischen Oberstufe bekannt (Rechnen mit Vektoren u. Determinanten, Deutung von Skalarprodukt, Kreuzprodukt, Darstellung v. Geraden/Ebenen, Satz d. Pythagoras, Cosinus-Satz, Satz d. Thales, Projektionen/Einbettungen) Grundlegende Datenstrukturen (Bäume, binäre Bäume, Suchbäume, balancierte binäre Bäume, Verkettete Listen) Definition eines Graphen		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Studierende können die grundlegenden Begriffe der Rechnergestützten Geometrie benennen, mathematisch exakt beschreiben und anhand von Beispielen erklären.</p> <p>Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.</p> <p><i>Fertigkeiten</i> Studierende können Aufgabenstellungen aus der Rechnergestützten Geometrie mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.</p>		
Personale Kompetenzen	<p><i>Sozialkompetenz</i> Studierende sind in der Lage, mit den Teilnehmerinnen und Teilnehmer ihre eigenen algorithmischen Vorschläge zur Lösung der vorgestellten Probleme zu erörtern. Studierende sind in der Lage, in Teams zusammenzuarbeiten und beherrschen die Mathematik als gemeinsame Sprache.</p> <p><i>Selbstständigkeit</i> Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.</p>		
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 II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Computer Science: Vertiefung Computermathematik: Wahlpflicht		

Lehrveranstaltung L0393: Rechnergestützte Geometrie	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Dr. Prashant Batra
Sprachen	DE
Zeitraum	WiSe
Inhalt	<p>Konstruktion einer konvexen Hülle von n Punkten</p> <p>Triangulierung eines schlichten Polygons</p> <p>Konstruktion einer Delaunay-Triangulation, eines Voronoi-Diagramms</p> <p>Algorithmen und Datenstrukturen zum Bestimmen eines Arrangements, eines Ham-Sandwich-Cuts. des Schnitts von Halbebenen, der Optimierung eines linearen Funktionals.</p> <p>Effiziente Bestimmung aller Schnittpunkte von (orthogonalen) Streckensegmenten</p> <p>Approximative Berechnung des Durchmessers einer Punktmenge</p> <p>Inkrementelle randomisierte Algorithmen</p> <p>Grundlagen der Gitterpunktlehre, LLL-Algorithmus und Anwendungen in der ganzzahligen Optimierung</p> <p>Grundlagen der Bewegungsplanung</p>
Literatur	<p>Computational Geometry Algorithms and Applications Authors:</p> <ul style="list-style-type: none"> • Prof. Dr. Mark de Berg, • Dr. Otfried Cheong, • Dr. Marc van Kreveld, • Prof. Dr. Mark Overmars <p>Springer e-Book: http://dx.doi.org/10.1007/978-3-540-77974-2</p> <p style="text-align: right;">Algorithmische Geometrie : Grundlagen, Methoden, Anwendungen / Rolf Klein Klein, Rolf 2., vollst. überarb. Aufl. Berlin [u.a.] : Springer, 2005 XI, 392 S. : graph. Darst.</p> <p>Verfasser: Ausgabe: Erschienen: Umfang:</p> <p>Springer e-Book: http://dx.doi.org/10.1007/3-540-27619-X</p> <p>O'Rourke, Joseph Computational geometry in C. (English) Zbl 0816.68124 Cambridge: Univ. Press. ix, 346 p. \$ 24.95; £16.95 /sc; \$ 59.95; £35.00 /hc (1994). ISBN: 0-521-44034-3 ; 0-521-44592-2</p> <p style="text-align: right;">Computational geometry : an introduction / Franco P. Preparata; Michael Ian Shamos Preparata, Franco P. ; Shamos, Michael Ian Corr. and expanded 2. printing. New York [u.a.] : Springer, 1988 XIV, 398 S. : graph. Darst. Texts and monographs in computer science 3-540-96131-3 0-387-96131-3</p> <p>Devadoss, Satyan L.; O'Rourke, Joseph Discrete and computational geometry. (English) Zbl 1232.52001 Princeton, NJ: Princeton University Press (ISBN 978-0-691-14553-2/hbk; 978-1-400-83898-1/ebook). xi, 255 p. ISBN: 978-3-540-77973-5 (Print) 978-3-540-77974-2 (Online)</p>

Lehrveranstaltung L0394: Rechnergestützte Geometrie	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Dr. Prashant Batra
Sprachen	DE
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0941: Kombinatorische Strukturen und Algorithmen			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Kombinatorische Strukturen und Algorithmen (L1100)	Vorlesung	3	4
Kombinatorische Strukturen und Algorithmen (L1101)	Gruppenübung	1	2
Modulverantwortlicher	Prof. Anusch Taraz		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Mathematik I + II • Diskrete Algebraische Strukturen • Graphentheorie und Optimierung 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<ul style="list-style-type: none"> • Studierende können die grundlegenden Begriffe der Kombinatorik und Algorithmik 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. 		
<i>Wissen</i>			
<i>Fertigkeiten</i>			
Personale Kompetenzen			
<i>Sozialkompetenz</i>	<ul style="list-style-type: none"> • Studierende sind in der Lage, in Teams zusammenzuarbeiten und beherrschen die Mathematik als gemeinsame Sprache. • Sie können dabei insbesondere neue Konzepte adressatengerecht kommunizieren und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen. 		
<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> • Studierende können eigenständig ihr Verständnis komplexer 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 zielgerichtet 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 Computer- und Software-Engineering: Wahlpflicht Computer Science: Vertiefung Computermathematik: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht		

Lehrveranstaltung L1100: Kombinatorische Strukturen und Algorithmen	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Anusch Taraz, Dr. Dennis Clemens
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> • Zählprobleme • Strukturelle Graphentheorie • Analyse von Algorithmen • Extremale Kombinatorik • Zufällige diskrete Strukturen
Literatur	<ul style="list-style-type: none"> • M. Aigner: Diskrete Mathematik, Vieweg, 6. Aufl., 2006 • J. Matoušek & J. Nešetřil: Diskrete Mathematik - Eine Entdeckungsreise, Springer, 2007 • A. Steger: Diskrete Strukturen - Band 1: Kombinatorik, Graphentheorie, Algebra, Springer, 2. Aufl. 2007 • A. Taraz: Diskrete Mathematik, Birkhäuser, 2012.

Lehrveranstaltung L1101: Kombinatorische Strukturen und Algorithmen	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Anusch Taraz
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1242: Quantenmechanik für Studierende der Ingenieurwissenschaften			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Quantenmechanik für Studierende der Ingenieurwissenschaften (L1686)		Vorlesung	2 3
Quantenmechanik für Studierende der Ingenieurwissenschaften (L1688)		Gruppenübung	2 3
Modulverantwortlicher	Prof. Wolfgang Hansen		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Kenntnisse in Physik insbesondere im Bereich Optik und Wellenlehre; • Kenntnisse in der Mathematik, insbesondere lineare Algebra, Vektorrechnung, komplexe Zahlen und Fourierentwicklung 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Die Studierenden können grundlegende Begriffe und Prinzipien der Quantenmechanik beschreiben und erläutern. Sie kennen Gemeinsamkeiten und Unterschiede zur klassischen Physik und wissen, in welchen Situationen quantenmechanische Effekte erwartet werden können.		
<i>Wissen</i>			
<i>Fertigkeiten</i>	Die Studierenden sind in der Lage, quantenmechanische Konzepte und Methoden auf einfache Probleme anzuwenden. Sie sind umgekehrt auch in der Lage, die Voraussetzungen und Prinzipien einfacher Anwendungen der Quantenmechanik in elektrooptischen Bauelementen nachzuvollziehen.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Die Studierenden diskutieren den Vorlesungsstoff und präsentieren Lösungen einfacher quantenmechanischer Probleme in Kleingruppen während der Übungen.		
<i>Selbstständigkeit</i>	Die Studierenden sind in der Lage selbstständig einfache Lösungswege zu quantenmechanische Problemen zu erarbeiten. Sie sind so weit mit Konzepten der Quantenmechanik vertraut, dass sie sich selbstständig Literatur zu komplexeren Fragestellungen mit quantenmechanischem Hintergrund erarbeiten können.		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Verpflichtend Bonus	Art der Studienleistung	Beschreibung
	Nein Keiner	Schriftliche Ausarbeitung	optionale Vorlage von selbst ausgearbeiteten Lösungen zu den Übungen
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung Computermathematik: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Elektrotechnik: Kernqualifikation: Wahlpflicht		

Lehrveranstaltung L1686: Quantenmechanik für Studierende der Ingenieurwissenschaften	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Wolfgang Hansen
Sprachen	DE
Zeitraum	WiSe
Inhalt	Diese Veranstaltung führt in grundlegende Konzepte, Methoden und Begriffe der Quantenmechanik ein, die in den Materialwissenschaften wichtig sind. Anwendungen werden anhand konkreter Beispiele aus dem Bereich elektronischer und optischer Bauelemente diskutiert. Zentrale Begriffe und Themen sind: Schrödinger-Gleichung, Wellenfunktionen, Operatoren, Eigenzustände, Eigenwerte, Quantentöpfe, harmonischer Oszillator, Tunnelprozesse, resonante Tunneldiode, Bandstruktur, Zustandsdichte, Besetzungsverteilung, Zener-Diode, stationäre Störungsrechnung am Beispiel des Quantum Confined Stark Effekts, Fermis Goldene Regel und Übergangsmatrixelemente, Heterostrukturlaser, Quantenkaskadenlaser, Vielteilchensysteme, Moleküle und Austauschwechselwirkung, Quantenbits und Quantenkryptographie
Literatur	<ul style="list-style-type: none"> • David J. Griffiths: "Quantenmechanik, eine Einführung", Pearson (2012), ISBN 978-3-8632-6514-4. • David K. Ferry: "Quantum Mechanics", IOP Publishing (1995), ISBN 0-7503-0327-1 (hbk) bzw. 0-7503-0328-X (pbk). • M. Jaros: " Physics and Applications of Semiconductor Microstructures ", Clarendon Press (1989), ISBN: 0-19-851994-X bzw. 0-19-853927-4 (Pbk). • Randy Harris, "Modernes Physik Lehr- und Übungsbuch", 2. aktualisierte Auflage, Kapitel 3-10, Pearson (2013), ISBN 978-3-86894-115-9. • Michael A Nielsen and Isaac L. Chuang: "Quantum Computation and Quantum Information", 10. Auflage, Cambridge University Press (2011), ISBN: 1107002176 9781107002173. • Hiroyuki Sagawa and Nobuaki Yoshida: "Fundamentals of Quantum Information", World Scientific Publishing (2010), ISBN-13: 978-9814324236.

Lehrveranstaltung L1688: Quantenmechanik für Studierende der Ingenieurwissenschaften	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Wolfgang Hansen
Sprachen	DE
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1592: Statistik				
Lehrveranstaltungen				
Titel		Typ	SWS	LP
Statistik (L2430)		Vorlesung	3	4
Statistik (L2431)		Gruppenübung	1	2
Modulverantwortlicher	Prof. Matthias Schulte			
Zulassungsvoraussetzungen	Keine			
Empfohlene Vorkenntnisse	Stochastik (oder eine vergleichbare Lehrveranstaltung)			
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht			
Fachkompetenz <i>Wissen</i>	<ul style="list-style-type: none"> Studierende können die grundlegenden Begriffe der Statistik benennen und anhand von Beispielen erklären. Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern. 			
<i>Fertigkeiten</i>	<ul style="list-style-type: none"> Studierende können statistische Probleme mit Hilfe der kennengelernten Konzepte mathematisch modellieren und mit den erlernten Methoden lösen. Hierfür können sie die statische Software R einsetzen. 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. 			
Personale Kompetenzen <i>Sozialkompetenz</i>	<ul style="list-style-type: none"> Studierende können in heterogen zusammengesetzten Teams (z.B. an Hausaufgaben) zusammenarbeiten und ihre Ergebnisse vor der Gruppe präsentieren. Sie können sich dabei insbesondere gegenseitig neue Konzepte erklären und anhand von Beispielen das Verständnis der Mitstudierenden überprüfen und vertiefen. 			
<i>Selbstständigkeit</i>	<ul style="list-style-type: none"> Studierende können eigenständig ihr Verständnis mathematischer Konzepte überprüfen, noch offene Fragen auf den Punkt bringen und sich gegebenenfalls gezielt Hilfe holen. Studierende können ihr Wissen mit den Inhalten anderer Veranstaltungen in Verbindung bringen. 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	Klausur			
Prüfungsdauer und -umfang	90 min			
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Pflicht Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht Wirtschaftsingenieurwesen - Fachrichtung Logistik und Mobilität: Vertiefung Informationstechnologie: Wahlpflicht			

Lehrveranstaltung L2430: Statistik	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Matthias Schulte
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	<ul style="list-style-type: none"> Punktschätzer Konfidenzintervalle Hypothesentests Nichtparametrische Statistik Lineare Regression Zeitreihenanalyse Statistische Software (R)
Literatur	

Lehrveranstaltung L2431: Statistik	
Typ	Gruppenübung
SWS	1
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 46, Präsenzstudium 14
Dozenten	Prof. Matthias Schulte
Sprachen	DE/EN
Zeitraum	WiSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0634: Einführung in Medizintechnische Systeme			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Einführung in Medizintechnische Systeme (L0342)		Vorlesung	2 3
Einführung in Medizintechnische Systeme (L0343)		Projektseminar	2 2
Einführung in Medizintechnische Systeme (L1876)		Hörsaalübung	1 1
Modulverantwortlicher	Prof. Alexander Schläefer		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Grundlagen Mathematik (Algebra, Analysis) Grundlagen Stochastik Grundlagen Programmierung, R/Matlab		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz			
<i>Wissen</i>	Die Studierenden können Funktionsprinzipien ausgewählter medizintechnischer Systeme (beispielsweise bildgebende Systeme, Assistenzsysteme im OP, medizintechnische Informationssysteme) erklären. Sie können einen Überblick über regulatorische Rahmenbedingungen und Standards in der Medizintechnik geben.		
<i>Fertigkeiten</i>	Die Studierenden sind in der Lage, die Funktion eines medizintechnischen Systems im Anwendungskontext zu bewerten.		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Die Studierenden können in Gruppen ein medizintechnisches Thema als Projekt beschreiben, in Teilaufgaben untergliedern und gemeinsam bearbeiten.		
<i>Selbstständigkeit</i>	Die Studierenden können ihren Wissensstand einschätzen und ihre Arbeitsergebnisse dokumentieren. Sie können die erzielten Ergebnisse kritisch bewerten und in geeigneter Weise präsentieren.		
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	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Medizingenieurwesen: Pflicht Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Wahlpflicht Elektrotechnik: Kernqualifikation: Wahlpflicht Engineering Science: Vertiefung Medizingenieurwesen: Pflicht General Engineering Science (7 Semester): Vertiefung Medizingenieurwesen: Pflicht Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht Medizingenieurwesen: Vertiefung Künstliche Organe und Regenerative Medizin: Wahlpflicht Medizingenieurwesen: Vertiefung Implantate und Endoprothesen: Wahlpflicht Medizingenieurwesen: Vertiefung Medizin- und Regelungstechnik: Wahlpflicht Medizingenieurwesen: Vertiefung Management und Administration: Wahlpflicht Technomathematik: Vertiefung III. Ingenieurwissenschaften: Wahlpflicht		

Lehrveranstaltung L0342: Einführung in Medizintechnische Systeme	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Alexander Schläefer
Sprachen	DE
Zeitraum	SoSe
Inhalt	- Bildgebende Systeme - Assistenzsysteme im OP - Medizintechnische Sensorsysteme - Medizintechnische Informationssysteme - Regulatorische Rahmenbedingungen - Standards in der Medizintechnik Durch problembasiertes Lernen erfolgt die Vertiefung der Methoden aus der Vorlesung. Dies erfolgt in Form von Gruppenarbeit.
Literatur	Wird in der Veranstaltung bekannt gegeben.

Lehrveranstaltung L0343: Einführung in Medizintechnische Systeme	
Typ	Projektseminar
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Alexander Schlaefer
Sprachen	DE
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Lehrveranstaltung L1876: Einführung in Medizintechnische Systeme	
Typ	Hörsaalübung
SWS	1
LP	1
Arbeitsaufwand in Stunden	Eigenstudium 16, Präsenzstudium 14
Dozenten	Prof. Alexander Schlaefer
Sprachen	DE
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> - Bildgebende Systeme - Assistenzsysteme im OP - Medizintechnische Sensorsysteme - Medizintechnische Informationssysteme - Regulatorische Rahmenbedingungen - Standards in der Medizintechnik <p>Durch problembasiertes Lernen erfolgt die Vertiefung der Methoden aus der Vorlesung. Dies erfolgt in Form von Gruppenarbeit.</p>
Literatur	Wird in der Veranstaltung bekannt gegeben.

Modul M0668: Algebraische Methoden in der Regelungstechnik			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Algebraische Methoden in der Regelungstechnik (L0428)		Vorlesung	2 4
Algebraische Methoden in der Regelungstechnik (L0429)		Gruppenübung	2 2
Modulverantwortlicher	Dr. Prashant Batra		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Mathe I-III (Reelle Analysis, Lineare Algebra,) und entweder: Einführung in die Regelungstechnik (Beschreibung u. gewünschte Eigenschaften von Systemen, Zeitbereich/Frequenzbereich) oder: Diskrete Mathematik (Gruppen, Ringe, Ideale, Körper, Euklidischer Algorithmus)		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	Studierende können <ul style="list-style-type: none"> • Input-Output-Systeme polynomial beschreiben, • Faktorisierungsansätze für Übertragungsfunktionen erklären, • Stabilisierungsbedingungen für Systeme in coprimer stabiler Faktorisierung benennen. 		
<i>Wissen</i>			
<i>Fertigkeiten</i>	Studierende sind in der Lage <ul style="list-style-type: none"> • eine Synthese stabiler Regelkreise durchzuführen, • geeignete Analyse und Synthesemethoden zur Beschreibung aller stabilen Regelkreise anzuwenden sowie • die Erfüllung vorgegebener Leistungsmaße sicher zu stellen. 		
Personale Kompetenzen			
<i>Sozialkompetenz</i>	Die Studierenden sind nach Abschluss des Moduls in der Lage, fachspezifische Aufgaben alleine zu bearbeiten und die Resultate geeignet zu präsentieren.		
<i>Selbstständigkeit</i>	Die Studierenden können ihren Wissensstand mit Hilfe klausurnaher Aufgaben kontinuierlich überprüfen und auf dieser Basis ihre Lernprozesse steuern.		
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 Computermathematik: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Technomathematik: Vertiefung II. Informatik: Wahlpflicht		

Lehrveranstaltung L0428: Algebraische Methoden in der Regelungstechnik	
Typ	Vorlesung
SWS	2
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 92, Präsenzstudium 28
Dozenten	Dr. Prashant Batra
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<p>- Algebraische Methoden der Regelungstechnik, polynomialer Ansatz, Faktorisierungsbeschreibung</p> <p>- Beschreibung 1-dimensionaler Regelsysteme, Synthese von (minimalen) Regelsystemen durch algebraische Interpolationsmethoden,</p> <p>- Simultane Stabilisierbarkeit</p> <p>- Parametrisierung sämtlicher stabilisierenden Regler</p> <p>- Reglerentwurf bei Polvorgabe</p> <p>- Berücksichtigung von Systemeigenschaften: Störanfälligkeit, Sensitivität.</p> <p>- Polynomiale Matrizen, Beschreibung durch Links-Faktorisierungen.</p> <p>- Euklidischer Algorithmus u. Diophantische Gleichungen über Ringen</p> <p>- Smith-McMillan Normal Form</p> <p>- Synthese von Mehrgrößensystemen durch polynomiale Methoden</p>
Literatur	<ul style="list-style-type: none"> • Vidyasagar, M.: Control system synthesis: a factorization approach. The MIT Press, Cambridge/Mass. - London, 1985. • Vardulakis, A.I.G.: Linear multivariable control. Algebraic analysis and synthesis methods, John Wiley & Sons, Chichester, UK, 1991. • Chen, Chi-Tsong: Analog and digital control system design. Transfer-function, state-space, and algebraic methods. Oxford Univ. Press, 1995. • Kučera, V.: Analysis and Design of Discrete Linear Control Systems. Praha: Academia, 1991.

Lehrveranstaltung L0429: Algebraische Methoden in der Regelungstechnik	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Dr. Prashant Batra
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M1269: Labor Cyber-Physical Systems			
Lehrveranstaltungen			
Titel	Labor Cyber-Physical Systems (L1740)	Typ	Projekt-/problembasierte Lehrveranstaltung
		SWS	4
		LP	6
Modulverantwortlicher	Prof. Heiko Falk		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Modul "Eingebettete Systeme"		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i></p> <p>Cyber-Physical Systems (CPS) stehen über Sensoren, A/D- und D/A-Wandler und Aktoren in enger Verbindung mit ihrer Umgebung. Wegen der besonderen Einsatzgebiete kommen hier hochgradig spezialisierte Sensoren, Prozessoren und Aktoren zum Einsatz, die applikationsspezifisch auf ihr jeweiliges Einsatzgebiet ausgerichtet sind. Dementsprechend existiert - im Gegensatz zum klassischen Software Engineering - eine Vielzahl unterschiedlicher Techniken zur Spezifikation von CPS.</p> <p>In Form von rechnergestützten Versuchen mit Roboterbausätzen werden in dieser Veranstaltung die Grundzüge der Spezifikation und Modellierung von CPS vermittelt. Das Labor behandelt die Einführung in diese Systeme (Begriffsbildung, charakteristische Eigenschaften) und deren Spezifikationsprachen (models of computation, hierarchische Zustandsautomaten, Datenfluss-Modelle, Petri-Netze, imperative Techniken). Da CPS häufig Steuerungs- und Regelungsaufgaben erfüllen, wird das Labor praxisnah einfache Anwendungen aus der Regelungstechnik vermitteln. Die Versuche nutzen gängige Spezifikationswerkzeuge (MATLAB/Simulink, LabVIEW, NXC), um hiermit Cyber-Physical Systems zu modellieren, die über Sensoren und Aktoren mit ihrer Umwelt interagieren.</p> <p><i>Fertigkeiten</i></p> <p>Nach erfolgreichem Besuch der Veranstaltung sind die Studierenden in der Lage, einfache CPS zu entwickeln. Sie können Wechselwirkungen zwischen einem CPS und dessen umgebenden Prozessen beurteilen, der sich aus dem Kreislauf zwischen physikalischer Umwelt, Sensor, A/D-Wandler, digitalem Prozessor, D/A-Wandler und Aktor ergibt. Die Veranstaltung versetzt die Studierenden in die Lage, Modellierungstechniken miteinander vergleichen, deren Vor- und Nachteile abwägen, und geeignete Techniken zur Systementwicklung einsetzen zu können. Sie erwerben die Fähigkeit, diese Techniken im Rahmen konkreter praktischer Aufgabenstellungen anzuwenden. Sie haben erste Erfahrungen im hardwarenahen Software-Entwurf, im Umgang mit industrierelevanten Spezifikationswerkzeugen und im Entwurf einfacher Regelungssysteme erworben.</p> <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i></p> <p>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.</p> <p><i>Selbstständigkeit</i></p> <p>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.</p>		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Schriftliche Ausarbeitung		
Prüfungsdauer und -umfang	Durchführung und Beschreibung sämtlicher Versuche		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Vertiefung Informatik: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Computer Science: Vertiefung Computer- und Software-Engineering: Wahlpflicht General Engineering Science (7 Semester): Vertiefung Informatik: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht Mechatronics: Vertiefung Systementwurf: Wahlpflicht Mechatronics: Technischer Ergänzungskurs: Wahlpflicht		

Lehrveranstaltung L1740: Labor Cyber-Physical Systems	
Typ	Projekt-/problembasierte Lehrveranstaltung
SWS	4
LP	6
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56
Dozenten	Prof. Heiko Falk
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Versuch 1: Programmieren in NXC • Versuch 2: Programmierung des Roboters mit Matlab/Simulink • Programmierung des Roboters in LabVIEW
Literatur	<ul style="list-style-type: none"> • Peter Marwedel. Embedded System Design - Embedded System Foundations of Cyber-Physical Systems. 2nd Edition, Springer, 2012. • Begleitende Foliensätze

Modul M0715: Solvers for Sparse Linear Systems			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Löser für schwachbesetzte lineare Gleichungssysteme (L0583)		Vorlesung	2 3
Löser für schwachbesetzte lineare Gleichungssysteme (L0584)		Gruppenübung	2 3
Modulverantwortlicher	Prof. Sabine Le Borne		
Zulassungsvoraussetzungen	None		
Empfohlene Vorkenntnisse	<ul style="list-style-type: none"> • Mathematics I + II for Engineering students or Analysis & Lineare Algebra I + II for Technomathematicians • Programming experience in C 		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Students can</p> <ul style="list-style-type: none"> • list classical and modern iteration methods and their interrelationships, • repeat convergence statements for iterative methods, • explain aspects regarding the efficient implementation of iteration methods. <p><i>Fertigkeiten</i> Students are able to</p> <ul style="list-style-type: none"> • analyse, implement, test, and compare iterative methods, • analyse the convergence behaviour of iterative methods and, if applicable, compute convergence rates. <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i> Students are able to</p> <ul style="list-style-type: none"> • work together in heterogeneously composed teams (i.e., teams from different study programs and background knowledge), explain theoretical foundations and support each other with practical aspects regarding the implementation of algorithms. <p><i>Selbstständigkeit</i> Students are capable</p> <ul style="list-style-type: none"> • to assess whether the supporting theoretical and practical exercises are better solved individually or in a team, • to work on complex problems over an extended period of time, • to assess their individual progress and, if necessary, to ask questions and seek help. 		
Arbeitsaufwand in Stunden	Eigenstudium 124, Präsenzstudium 56		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Mündliche Prüfung		
Prüfungsdauer und -umfang	20 min		
Zuordnung zu folgenden Curricula	Computer Science: Vertiefung Computermathematik: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Wahlpflicht Informatik-Ingenieurwesen: Vertiefung II. Mathematik & Ingenieurwissenschaften: Wahlpflicht Technomathematik: Vertiefung I. Mathematik: Wahlpflicht		

Lehrveranstaltung L0583: Solvers for Sparse Linear Systems	
Typ	Vorlesung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne
Sprachen	EN
Zeitraum	SoSe
Inhalt	<ol style="list-style-type: none"> 1. Sparse systems: Orderings and storage formats, direct solvers 2. Classical methods: basic notions, convergence 3. Projection methods 4. Krylov space methods 5. Preconditioning (e.g. ILU) 6. Multigrid methods 7. Domain Decomposition Methods
Literatur	<ol style="list-style-type: none"> 1. Y. Saad. Iterative methods for sparse linear systems 2. M. Olshanskii, E. Tyrtshnikov. Iterative methods for linear systems: theory and applications

Lehrveranstaltung L0584: Solvers for Sparse Linear Systems	
Typ	Gruppenübung
SWS	2
LP	3
Arbeitsaufwand in Stunden	Eigenstudium 62, Präsenzstudium 28
Dozenten	Prof. Sabine Le Borne
Sprachen	EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Modul M0672: Signale und Systeme			
Lehrveranstaltungen			
Titel		Typ	SWS LP
Signale und Systeme (L0432)		Vorlesung	3 4
Signale und Systeme (L0433)		Gruppenübung	2 2
Modulverantwortlicher	Prof. Gerhard Bauch		
Zulassungsvoraussetzungen	Keine		
Empfohlene Vorkenntnisse	Mathematik 1-3 Das Modul führt in das Thema der Signal- und Systemtheorie ein. Sicherer Umgang mit grundlegenden mathematischen Methoden, wie sie in den Modulen Mathematik 1-3 vermittelt werden, wird erwartet. Darüber hinaus sind Vorkenntnisse in Grundlagen von Spektraltransformationen (Fourier-Reihe, Fourier-Transformation, Laplace-Transformation) zwar nützlich, aber keine Voraussetzung.		
Modulziele/ angestrebte Lernergebnisse	Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht		
Fachkompetenz	<p><i>Wissen</i> Die Studierenden können Signale und lineare zeitinvariante (LTI) Systeme im Sinne der Signal- und Systemtheorie klassifizieren und beschreiben. Sie beherrschen die grundlegenden Integraltransformationen zeitkontinuierlicher und zeitdiskreter deterministischer Signale und Systeme. Sie können deterministische Signale und Systeme in Zeit- und Bildbereich mathematisch beschreiben und analysieren. Sie verstehen elementare Operationen und Konzepte der Signalverarbeitung und können diese in Zeit- und Bildbereich beschreiben. Insbesondere verstehen Sie die mit dem Übergang vom zeitkontinuierlichen zum zeitdiskreten Signal bzw. System einhergehenden Effekte in Zeit- und Bildbereich.</p> <p><i>Fertigkeiten</i> Die Studierenden können deterministische Signale und lineare zeitinvariante Systeme mit den Methoden der Signal- und Systemtheorie beschreiben und analysieren. Sie können einfache Systeme hinsichtlich wichtiger Eigenschaften wie Betrags- und Phasenfrequenzgang, Stabilität, Linearität etc. analysieren und entwerfen. Sie können den Einfluß von LTI-Systemen auf die Signaleigenschaften in Zeit- und Frequenzbereich beurteilen.</p> <p>Personale Kompetenzen</p> <p><i>Sozialkompetenz</i> Die Studierenden können fachspezifische Aufgaben gemeinsam bearbeiten.</p> <p><i>Selbstständigkeit</i> Die Studierenden sind in der Lage, die notwendigen Informationen aus geeigneten Literaturquellen selbstständig zu beschaffen und in den Kontext der Vorlesung zu setzen. Sie können ihren Wissensstand mit Hilfe vorlesungsbegleitender Maßnahmen (klausurnahe Aufgaben, Software-Tools, Clicker-System) kontinuierlich überprüfen und auf dieser Basis ihre Lernprozesse steuern.</p>		
Arbeitsaufwand in Stunden	Eigenstudium 110, Präsenzstudium 70		
Leistungspunkte	6		
Studienleistung	Keine		
Prüfung	Klausur		
Prüfungsdauer und -umfang	90 min		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften (7 Semester): Kernqualifikation: Pflicht Computer Science: Kernqualifikation: Pflicht Computer Science: Vertiefung II. Mathematik und Ingenieurwissenschaften: Wahlpflicht Data Science: Kernqualifikation: Pflicht Elektrotechnik: Kernqualifikation: Pflicht Informatik-Ingenieurwesen: Kernqualifikation: Pflicht Maschinenbau: Vertiefung Mechatronik: Wahlpflicht Mechatronik: Kernqualifikation: Pflicht Technomathematik: Vertiefung III. Ingenieurwissenschaften: Wahlpflicht		

Lehrveranstaltung L0432: Signale und Systeme	
Typ	Vorlesung
SWS	3
LP	4
Arbeitsaufwand in Stunden	Eigenstudium 78, Präsenzstudium 42
Dozenten	Prof. Gerhard Bauch
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	<ul style="list-style-type: none"> • Einführung in die Signal- und Systemtheorie • Signale <ul style="list-style-type: none"> ◦ Klassifikation von Signalen <ul style="list-style-type: none"> ▪ Zeitkontinuierliche und zeitdiskrete Signale ▪ Analoge und digitale Signale ▪ Deterministische und zufällige Signale ◦ Beschreibung von LTI-Systemen durch Differentialgleichungen bzw. Differenzgleichungen ◦ Grundlegende Eigenschaften von Signalen und grundlegende Operationen ◦ Elementare Signale ◦ Distributionen ◦ Leistung und Energie von Signalen ◦ Korrelationsfunktionen deterministischer Signale

- Autokorrelationsfunktion
 - Kreuzkorrelationsfunktion
 - Orthogonale Signale
 - Anwendungen der Korrelation
- Lineare zeitinvariante Systeme (linear time-invariant (LTI) systems)
 - Linearität
 - Zeitinvarianz
 - Beschreibung von LTI-Systemen durch Impulsantwort und Übertragungsfunktion
 - Faltung
 - Faltung und Korrelation
 - Eigenschaften von LTI-Systemen
 - Kausale Systeme
 - Stabile Systeme
 - Gedächtnislose Systeme
- Fourier-Reihe und Fourier-Transformation
 - Fourier-Transformation zeitkontinuierlicher, zeitdiskreter, periodischer und nicht-periodischer Signale
 - Eigenschaften der Fourier-Transformation
 - Fourier-Transformation einiger elementarer Signale
 - Parsevalsches Theorem
- Analyse von LTI-Systemen und Signalen im Frequenzbereich
 - Übertragungsfunktion, Betragsfrequenzgang, Phasengang
 - Übertragungsfaktor, Dämpfung, Gewinn
 - Frequenzselektive und nicht-frequenzselektive LTI-Systeme
 - Bandbreite-Definitionen
 - Grundlegende Typen von Systemen (Filtern): Tiefpass, Hochpass, Bandpass, Bandsperre
 - Phasenlaufzeit und Gruppenlaufzeit
 - Linearphasige Systeme
 - Verzerrungsfreie Systeme
 - Spektralanalyse mit begrenztem Beobachtungsfenster: Leck-Effekt
- Laplace-Transformation
 - Zusammenhang von Fourier-Transformation und Laplace-Transformation
 - Eigenschaften der Laplace-Transformation
 - Laplace-Transformation einiger elementarer Signale
- Analyse von LTI-Systemen im s-Bereich
 - Übertragungsfunktion von LTI-Systemen
 - Zusammenhang von Laplace-Transformation, Betragsfrequenzgang und Phasengang
 - Analyse von LTI-Systemen mit Pol-Nullstellen-Diagrammen
 - Allpass-Filter
 - Minimalphasige, maximalphasige und gemischtphasige Filter
 - Stabile Systeme
- Abtastung
 - Abtasttheorem
 - Rekonstruktion des zeitkontinuierlichen Signals in Frequenz- und Zeitbereich
 - Überabtastung
 - Aliasing
 - Abtastung mit Pulsen endlicher Dauer, Sample and Hold
 - Dezimierung und Interpolation
- Zeitdiskrete Fourier-Transformation (Discrete-Time Fourier Transform (DTFT))
 - Zusammenhang zwischen Fourier-Transformation und DTFT
 - Eigenschaften der DTFT
- Diskrete Fourier-Transformation (Discrete Fourier Transform (DFT))
 - Zusammenhang zwischen DTFT und DFT
 - Zyklische Eigenschaften der DFT
 - DFT-Matrix
 - Zero-Padding
 - Zyklische Faltung
 - Schnelle Fourier-Transformation (Fast Fourier Transform (FFT))
 - Anwendung der DFT: Orthogonal Frequency Division Multiplex (OFDM)
- Z-Transformation
 - Zusammenhang zwischen Laplace-Transformation, DTFT, und z-Transformation
 - Eigenschaften der z-Transformation
 - Z-transform einiger elementarer zeitdiskreter Signale
- Zeitdiskrete Systeme, Digitale Filter
 - FIR und IIR Filter
 - Z-Transformation digitaler Filter
 - Analyse zeitdiskreter Systeme mit Pol-Nullstellen-Diagrammen im z-Bereich
 - Stabilität
 - Allpass-Filter
 - Minimalphasige, maximalphasige und gemischtphasige Filter
 - Linearphasige Filter

Literatur

- T. Frey , M. Bossert , Signal- und Systemtheorie, B.G. Teubner Verlag 2004
- K. Kammeyer, K. Kroschel, Digitale Signalverarbeitung, Teubner Verlag.

	<ul style="list-style-type: none"> • B. Girod ,R. Rabensteiner , A. Stenger , Einführung in die Systemtheorie, B.G. Teubner, Stuttgart, 1997 • J.R. Ohm, H.D. Lüke , Signalübertragung, Springer-Verlag 8. Auflage, 2002 • S. Haykin, B. van Veen: Signals and systems. Wiley. • Oppenheim, A.S. Willsky: Signals and Systems. Pearson. • Oppenheim, R. W. Schafer: Discrete-time signal processing. Pearson.
--	--

Lehrveranstaltung L0433: Signale und Systeme	
Typ	Gruppenübung
SWS	2
LP	2
Arbeitsaufwand in Stunden	Eigenstudium 32, Präsenzstudium 28
Dozenten	Prof. Gerhard Bauch
Sprachen	DE/EN
Zeitraum	SoSe
Inhalt	Siehe korrespondierende Vorlesung
Literatur	Siehe korrespondierende Vorlesung

Fachmodule der Vertiefung III. Fachspezifische Fokussierung

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

Modul M1562: Technischer Ergänzungskurs I für CSBS			
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 III. Fachspezifische Fokussierung: Wahlpflicht		

Modul M1568: Technischer Ergänzungskurs II für CSBS			
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 III. Fachspezifische Fokussierung: Wahlpflicht		

Thesis

Modul M-001: Bachelorarbeit			
Lehrveranstaltungen			
Titel	Typ	SWS	LP
Modulverantwortlicher	Professoren der TUHH		
Zulassungsvoraussetzungen	<ul style="list-style-type: none"> Laut ASPO § 21 (1): Es müssen mindestens 126 Leistungspunkte im Studiengang erworben worden sein. Über Ausnahmen entscheidet der Prüfungsausschuss. 		
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>	<ul style="list-style-type: none"> Studierende können die wichtigsten wissenschaftlichen Grundlagen ihres Studienfaches (Fakten, Theorien und Methoden) problembezogen auswählen, darstellen und nötigenfalls kritisch diskutieren. Die Studierenden können ausgehend von ihrem fachlichen Grundlagenwissen anlassbezogen auch weiterführendes fachliches Wissen erschließen und verknüpfen. Die Studierenden können zu einem ausgewählten Thema ihres Faches einen Forschungsstand darstellen. Die Studierenden können das im Studium vermittelte Grundwissen ihres Studienfaches zielgerichtet zur Lösung fachlicher Probleme einsetzen. Die Studierenden können mit Hilfe der im Studium erlernten Methoden Fragestellungen analysieren, fachliche Sachverhalte entscheiden und Lösungen entwickeln. Die Studierenden können zu den Ergebnissen ihrer eigenen Forschungsarbeit kritisch aus einer Fachperspektive Stellung beziehen. Studierende können eine wissenschaftliche Fragestellung für ein Fachpublikum sowohl schriftlich als auch mündlich strukturiert, verständlich und sachlich richtig darstellen. Studierende können in einer Fachdiskussion auf Fragen eingehen und sie in adressatengerechter Weise beantworten. Sie können dabei eigene Einschätzungen und Standpunkte überzeugend vertreten. Studierende können einen umfangreichen Arbeitsprozess zeitlich strukturieren und eine Fragestellung in vorgegebener Frist bearbeiten. Studierende können notwendiges Wissen und Material zur Bearbeitung eines wissenschaftlichen Problems identifizieren, erschließen und verknüpfen. Studierende können die wesentlichen Techniken des wissenschaftlichen Arbeitens in einer eigenen Forschungsarbeit anwenden. 		
Arbeitsaufwand in Stunden	Eigenstudium 360, Präsenzstudium 0		
Leistungspunkte	12		
Studienleistung	Keine		
Prüfung	Abschlussarbeit		
Prüfungsdauer und -umfang	laut ASPO		
Zuordnung zu folgenden Curricula	Allgemeine Ingenieurwissenschaften: Abschlussarbeit: Pflicht Allgemeine Ingenieurwissenschaften (7 Semester): Abschlussarbeit: Pflicht Bau- und Umweltingenieurwesen: Abschlussarbeit: Pflicht Bioverfahrenstechnik: Abschlussarbeit: Pflicht Computer Science: Abschlussarbeit: Pflicht Data Science: Abschlussarbeit: Pflicht Digitaler Maschinenbau: Abschlussarbeit: Pflicht Elektrotechnik: Abschlussarbeit: Pflicht Energie- und Umwelttechnik: Abschlussarbeit: Pflicht Engineering Science: Abschlussarbeit: Pflicht General Engineering Science: Abschlussarbeit: Pflicht General Engineering Science (7 Semester): Abschlussarbeit: Pflicht Green Technologies: Energie, Wasser, Klima: Abschlussarbeit: Pflicht Informatik-Ingenieurwesen: Abschlussarbeit: Pflicht Logistik und Mobilität: Abschlussarbeit: Pflicht Maschinenbau: Abschlussarbeit: Pflicht Mechatronik: Abschlussarbeit: Pflicht Schiffbau: Abschlussarbeit: Pflicht Technomathematik: Abschlussarbeit: Pflicht Teilstudiengang Lehramt Elektrotechnik-Informationstechnik: Abschlussarbeit: Pflicht Teilstudiengang Lehramt Metalltechnik: Abschlussarbeit: Pflicht Verfahrenstechnik: Abschlussarbeit: Pflicht		

