



Modulhandbuch

Bachelor of Science Computer Science

Wintersemester 2015

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Studiengangsbeschreibung	3
Fachmodule der Kernqualifikation	5
Modul: Diskrete Algebraische Strukturen	5
Modul: Prozedurale Programmierung	7
Modul: Nichttechnische Ergänzungskurse im Bachelor	10
Modul: Functional Programming	12
Modul: Linear Algebra	14
Modul: Objektorientierte Programmierung, Algorithmen und Datenstrukturen	16
Modul: Logic, Automata and Formal Languages	18
Modul: Software Engineering	20
Modul: Mathematical Analysis	22
Modul: Grundlagen der Betriebswirtschaftslehre	24
Modul: Technische Informatik	27
Modul: Computernetworks and Internet Security	30
Modul: Mathematik III	32
Modul: Introduction to Information Security	35
Modul: Berechenbarkeit und Komplexität	37
Modul: Signale und Systeme	39
Modul: Stochastics	41
Modul: Graphentheorie und Optimierung	43
Modul: Betriebssysteme	45
Modul: Seminare Informatik und Mathematik	47
Modul: Software-Fachpraktikum	49
Fachmodule der Vertiefung Computational Mathematics	50
Modul: Rechnergestützte Geometrie	50
Modul: Numerische Mathematik I	52
Modul: Grundlagen der Regelungstechnik	54
Modul: Numerik und Computer Algebra	57
Modul: Kombinatorische Strukturen und Algorithmen	59
Modul: Algebraische Methoden in der Regelungstechnik	61
Modul: Löser für schwachbesetzte lineare Gleichungssysteme	63
Modul: Mathematik IV	65
Fachmodule der Vertiefung Computer Engineering	68
Modul: Grundlagen der Informatik	68
Modul: Einführung in die Nachrichtentechnik und ihre stochastischen Methoden	70
Modul: Messtechnik und Messdatenverarbeitung	72
Modul: Rechnerarchitektur	74
Modul: Verteilte Systeme	76
Modul: Quantenmechanik für Studierende der Ingenieurwissenschaften	78
Modul: Databases	80
Modul: Einführung in Medizintechnische Systeme	82
Modul: Compiler Construction	84
Modul: Application Security	86
Modul: Halbleiterschaltungstechnik	88
Modul: Eingebettete Systeme	90
Modul: Labor Cyber-Physical Systems	92
Thesis	94
Modul: Bachelorarbeit	94

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 vertieftes Wissen auch in Mathematik und 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, Grammatiken, graphentheoretische Netzwerke, Differentialgleichungen, Regelkreise, 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, Gradienten-basierte 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 einer verteilten Arrangierung 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.
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

Modulhandbuch - Bachelor of Science "Computer Science"

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 die 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 Bachelorstudiengangs Computer Science ist wie folgt gegliedert:

- Kernqualifikation: 21 Module, 132 Leistungspunkte (LP), 1. – 6. Semester
- Vertiefung: 6 Module, 36 LP, 1., 5. und 6. Semester
- Bachelorarbeit: 12 LP, 6. Semester

Damit ergibt sich ein Gesamtaufwand von 180 LP.

Die fachliche Lehre der Kernqualifikation erstreckt sich vorwiegend über das 1. bis 5. Semester.

Die Kernqualifikation beinhaltet neben Fachmodulen auch folgende Module:

- Grundlagen der Betriebswirtschaftslehre: 6 LP, 2. Semester
- Nichttechnische Ergänzungskurse im Bachelor: 6 LP, 1. – 6. Semester

Ebenfalls zur Kernqualifikation gehört das Software-Fachpraktikum (6 LP, 5. Semester).

In der Vertiefung werden fachliche Schlüsselqualifikationen erworben. Wählbar sind:

- Computerorientierte Mathematik oder
- Technische Informatik.

Die Studierenden belegen in einem der beiden Zweige Veranstaltungen in einem Umfang von 36 LP. In jedem Block werden

Veranstaltungen in einem Gesamtumfang von mindestens 48 LP vorgehalten, sodass ausreichend Wahlmöglichkeiten bestehen.

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

Fachmodule der Kernqualifikation

Modul: Diskrete Algebraische Strukturen

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Diskrete Algebraische Strukturen	Vorlesung	2
Diskrete Algebraische Strukturen	Gruppenübung	2

Modulverantwortlich:

Prof. Karl-Heinz Zimmermann

Zulassungsvoraussetzung:

Keine.

Empfohlene Vorkenntnisse:

Abiturkenntnisse in Mathematik.

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Wissen: Die Studierenden kennen

- 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 (Computeralgebrasysteme) zur Lösung von algebraischen oder kombinatorischen Aufgabenstellungen.

Fertigkeiten:

Fertigkeiten: Die Studierenden können

- 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 machen;
- einfache, im Kontext stehende mathematische Aussagen beweisen;
- einschlägige mathematische Software (Computeralgebrasysteme) zielgerichtet einsetzen.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachbüchern selbständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Technomathematik: Vertiefung Mathematik: Wahlpflicht

Lehrveranstaltung: Diskrete Algebraische Strukturen (Vorlesung)

Dozenten:

Prof. Karl-Heinz Zimmermann

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Literatur:

Lehrveranstaltung: Diskrete Algebraische Strukturen (Gruppenübung)

Dozenten:

Prof. Karl-Heinz Zimmermann

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Prozedurale Programmierung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Prozedurale Programmierung	Vorlesung	1
Prozedurale Programmierung	Gruppenübung	1
Prozedurale Programmierung	Laborpraktikum	2

Modulverantwortlich:

Prof. Siegfried Rump

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Elementare Handhabung eines PC
Elementare Mathematikkenntnisse

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden erwerben folgendes Wissen:

- Sie kennen elementare Sprachelemente der Programmiersprache C. Sie kennen die grundlegenden Datentypen und wissen um ihre Einsatzgebiete.
- Sie haben ein Verständnis davon, was die Aufgaben eines Compilers, des Präprozessors und der Entwicklungsumgebung sind und wie diese interagieren.
- Sie beherrschen die Einbindung und Verwendung externer Programm-Bibliotheken zur Erweiterung des Funktionsumfangs.
- Sie wissen, wie man Header-Dateien verwendet und Funktionsschnittstellen festlegt, um größere Programmierprojekte kreieren zu können.
- Sie haben ein Verständnis dafür, wie das implementierte Programm mit dem Betriebssystem interagiert. Dies befähigt Sie dazu, Programme zu entwickeln, welche Eingaben des Benutzers, Betriebseingaben oder auch entsprechende Dateien verarbeiten und gewünschte Ausgaben erzeugen.
- Sie haben mehrere Herangehensweisen zur Implementierung häufig verwendeter Algorithmen gelernt.

Fertigkeiten:

- Die Studierenden sind in der Lage, die Komplexität eines Algorithmus zu bewerten und eine effiziente Implementierung vorzunehmen.
- Die Studierenden können Algorithmen für eine Vielzahl von Funktionalitäten modellieren und programmieren. Zudem können Sie die Implementierung an eine vorgegebene API anpassen.

Personale Kompetenzen:

Sozialkompetenz:

Die Studierenden erwerben folgende Kompetenzen:

- Sie können in Kleingruppen Aufgaben gemeinsam lösen, Programmfehler analysieren und beheben und ihr erzieltes Ergebnis gemeinsam präsentieren.
- Sie können sich Sachverhalte direkt am Rechner durch einfaches Ausprobieren gegenseitig klar machen.
- Sie können in Kleingruppen gemeinsam eine Projektidee und -planung erarbeiten.
- Sie müssen den betreuenden Tutoren ihre eigenen Lösungsansätze verständlich kommunizieren und ihre Programme präsentieren.

Selbstständigkeit:

- Die Studierenden müssen in Einzeltestaten sowie einer abschließenden Prüfung ihre Programmierfertigkeiten unter Beweis stellen und selbstständig ihr erlerntes Wissen zur Lösung neuer Aufgabenstellungen anwenden.
- Die Studierenden haben die Möglichkeit, ihre erlernten Fähigkeiten beim Lösen einer Vielzahl von Präsenzaufgaben zu überprüfen.
- Zur effizienten Bearbeitung der Aufgaben des Praktikums teilen die Studierenden innerhalb ihrer Gruppen die Übungsaufgaben auf. Jeder Studierende muss zunächst selbstständig eine Teilaufgabe lösen.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht
Computer Science myTrack: Kernqualifikation: Pflicht
Elektrotechnik: Kernqualifikation: Pflicht

Modulhandbuch - Bachelor of Science "Computer Science"

Elektrotechnik myTrack: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht
Mechatronik: Kernqualifikation: Pflicht
Technomathematik: Kernqualifikation: Pflicht

Lehrveranstaltung: Prozedurale Programmierung (Vorlesung)

Dozenten:

Prof. Siegfried Rump

Sprachen:

DE

Zeitraum:

WS

Inhalt:

- elementare Datentypen (Integer, Gleitpunktformat, ASCII-Zeichen) und ihre Abhängigkeiten von der Architektur
- höhere Datentypen (Zeiger, Arrays, Strings, Strukturen, Listen)
- Operatoren (arithmetische Operationen, logische Operationen, Bit-Operationen)
- Kontrollflussstrukturen (bedingte Verzweigung, Schleifen, Sprünge)
- Präprozessor-Direktiven (Makros, bedingte Kompilierung, modulares Design)
- Funktionen (Funktionsdefinition/-interface, Rekursion, "call by value" versus "call by reference", Funktionszeiger)
- essentielle Standard-Bibliotheken und -Funktionen (stdio.h, stdlib.h, math.h, string.h, time.h)
- Dateikonzept, Streams
- einfache Algorithmen (Sortierfunktionen, Reihenentwicklung, gleichverteilte Permutation)
- Übungsprogramme zur Vertiefung der Programmierkenntnisse

Literatur:

Kernighan, Brian W (Ritchie, Dennis M.):

The C programming language

ISBN: 9780131103702

Upper Saddle River, NJ [u.a.] : Prentice Hall PTR, 2009

Sedgewick, Robert

Algorithms in C

ISBN: 0201316633

Reading, Mass. [u.a.] : Addison-Wesley, 2007

Kaiser, Ulrich (Kecher, Christoph.):

C/C++: Von den Grundlagen zur professionellen Programmierung

ISBN: 9783898428392

Bonn : Galileo Press, 2010

Wolf, Jürgen

C von A bis Z : das umfassende Handbuch

ISBN: 3836214113

Bonn : Galileo Press, 2009

Lehrveranstaltung: Prozedurale Programmierung (Gruppenübung)

Dozenten:

Prof. Siegfried Rump

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Prozedurale Programmierung (Laborpraktikum)

Dozenten:

Prof. Siegfried Rump

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Modulhandbuch - Bachelor of Science "Computer Science"

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modulverantwortlich:

Dagmar Richter

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

siehe jeweilige Veranstaltungsbeschreibung

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Der Studienbereich Nichttechnische Wahlpflichtfächer

vermittelt die in Hinblick auf das Ausbildungsprofil der TUHH nötigen Kompetenzen, die ingenieurwissenschaftliche Fachlehre fördern aber nicht abschließend behandeln kann: Eigenverantwortlichkeit, Selbstführung, Zusammenarbeit und fachliche wie personale Leitungsbefähigung der zukünftigen Ingenieurinnen und Ingenieure. Er setzt diese Ausbildungsziele in seiner **Lehrarchitektur**, den **Lehr-Lern-Arrangements**, den **Lehrbereichen** und durch Lehrangebote um, in denen sich Studierende wahlweise für **spezifische Kompetenzen** und ein **Kompetenzniveau** auf Bachelor- oder Masterebene qualifizieren können. Die Lehrangebote sind jeweils in einem Modulkatalog Nichttechnische Ergänzungskurse zusammengefasst.

Die Lehrarchitektur

besteht aus einem studiengangübergreifenden Pflichtstudienangebot. Durch dieses zentral konzipierte Lehrangebot wird die Profilierung der TUHH Ausbildung auch im „Nichttechnischen Studienbereich“ gewährleistet.

Die Lernarchitektur erfordert und übt eigenverantwortliche Bildungsplanung in Hinblick auf den individuellen Kompetenzaufbau ein und stellt dazu Orientierungswissen zu thematischen Schwerpunkten von Veranstaltungen bereit.

Das über den gesamten Studienverlauf begleitend studierbare Angebot kann ggf. in ein-zwei Semestern studiert werden. Angesichts der bekannten, individuellen Anpassungsprobleme beim Übergang von Schule zu Hochschule in den ersten Semestern und um individuell geplante Auslandsemester zu fördern, wird jedoch von einer Studienfixierung in konkreten Fachsemestern abgesehen.

Die Lehr-Lern-Arrangements

sehen für Studierende - nach B.Sc. und M.Sc. getrennt - ein semester- und fachübergreifendes voneinander Lernen vor. Der Umgang mit Interdisziplinarität und einer Vielfalt von Lernständen in Veranstaltungen wird eingeübt - und in spezifischen Veranstaltungen gezielt gefördert.

Die Lehrbereiche

basieren auf Forschungsergebnissen aus den wissenschaftlichen Disziplinen Kulturwissenschaften, Gesellschaftswissenschaften, Kunst, Geschichtswissenschaften, Kommunikationswissenschaften, Nachhaltigkeitsforschung und aus der Fachdidaktik der Ingenieurwissenschaften. Über alle Studiengänge hinweg besteht im Bachelorbereich zusätzlich ab Wintersemester 2014/15 das Angebot, gezielt Betriebswirtschaftliches und Gründungswissen aufzubauen. Das Lehrangebot wird durch soft skill und Fremdsprachkurse ergänzt. Hier werden insbesondere kommunikative Kompetenzen z.B. für Outgoing Engineers gezielt gefördert.

Das Kompetenzniveau

der Veranstaltungen in den Modulen der nichttechnischen Ergänzungskurse unterscheidet sich in Hinblick auf das zugrunde gelegte Ausbildungsziel: Diese Unterschiede spiegeln sich in den verwendeten Praxisbeispielen, in den - auf unterschiedliche berufliche Anwendungskontexte verweisende – Inhalten und im für M.Sc. stärker wissenschaftlich-theoretischen Abstraktionsniveau. Die Soft skills für Bachelor- und für Masterabsolventinnen/ Absolventen unterscheidet sich an Hand der im Berufsleben unterschiedlichen Positionen im Team und bei der Anleitung von Gruppen.

Fachkompetenz (Wissen)

Die Studierenden können

- ausgewählte Spezialgebiete 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 NTW-Bereich ist).

Fertigkeiten:

Die Studierenden können in ausgewählten Teilbereichen

- 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:

Sozialkompetenz:

Die Studierenden sind fähig ,

- in unterschiedlichem Ausmaß kooperativ zu lernen
- eigene Aufgabenstellungen in den o.g. Bereichen in adressatengerechter Weise in einer Partner- oder Gruppensituation zu präsentieren und zu analysieren,
- nichttechnische Fragestellungen einer Zuhörerschaft mit technischem Hintergrund verständlich darzustellen
- sich landessprachlich kompetent, kulturell angemessen und geschlechtersensibel auszudrücken (sofern dies der gewählte Schwerpunkt im NTW-Bereich ist)

Selbstständigkeit:

Die Studierenden sind in ausgewählten Bereichen in der Lage,

- die eigene Profession und Professionalität im Kontext der lebensweltlichen Anwendungsgebiete zu reflektieren,
- sich selbst und die eigenen Lernprozesse zu organisieren,
- Fragestellungen vor einem breiten Bildungshorizont zu reflektieren und verantwortlich zu entscheiden,
- sich in Bezug auf ein nichttechnisches Sachthema mündlich oder schriftlich kompetent auszudrücken.
- sich als unternehmerisches Subjekt zu organisieren, (sofern dies ein gewählter Schwerpunkt im NTW-Bereich ist).

Leistungspunkte:

6 LP

Arbeitsaufwand in Stunden:

Eigenstudium: 96, Präsenzstudium: 84

Lehrveranstaltungen:

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

Modul: Functional Programming

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Funktionales Programmieren	Vorlesung	2
Funktionales Programmieren	Hörsaalübung	2
Funktionales Programmieren	Laborpraktikum	2

Modulverantwortlich:

Prof. Sibylle Schupp

Zulassungsvoraussetzung:

None

Empfohlene Vorkenntnisse:

Discrete mathematics at high-school level

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

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.

Fertigkeiten:

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:

Sozialkompetenz:

Students practice peer programming with varying peers. They explain problems and solutions to their peer. They defend their programs orally. They communicate in English.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 96, Präsenzstudium: 84

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Functional Programming (Vorlesung)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

WS

Inhalt:

- 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

Modulhandbuch - Bachelor of Science "Computer Science"

- 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: Functional Programming (Hörsaalübung)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

WS

Inhalt:

- 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: Functional Programming (Laborpraktikum)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Linear Algebra

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Lineare Algebra	Vorlesung	4
Lineare Algebra	Hörsaalübung	2
Lineare Algebra	Gruppenübung	2

Modulverantwortlich:

Prof. Marko Lindner

Zulassungsvoraussetzung:

Empfohlene Vorkenntnisse:

School mathematics

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- Students can name the basic concepts in linear algebra. They are able to explain them using appropriate examples.
- Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.
- They know proof strategies and can reproduce them.

Fertigkeiten:

- Students can model problems in linear algebra with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.
- Students are able to discover and verify further logical connections between the concepts studied in the course.
- For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

8 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 128, Präsenzstudium: 112

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

General Engineering Science: Kernqualifikation: Pflicht

Lehrveranstaltung: Linear Algebra (Vorlesung)

Dozenten:

Dr. Francisco Javier Hoecker-Escuti

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Preliminaries

Vector spaces

Matrices and linear systems of equations

Scalar products and orthogonality

Basis transformation

Determinants

Eigen values

Literatur:

Strang: Linear Algebra
Beutelsbacher: Lineare Algebra

Lehrveranstaltung: Linear Algebra (Hörsaalübung)

Dozenten:

Dr. Francisco Javier Hoecker-Escuti

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Linear Algebra (Gruppenübung)

Dozenten:

Dr. Francisco Javier Hoecker-Escuti

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Objektorientierte Programmierung, Algorithmen und Datenstrukturen	Vorlesung	4
Objektorientierte Programmierung, Algorithmen und Datenstrukturen	Gruppenübung	1

Modulverantwortlich:

Prof. Rolf-Rainer Grigat

Zulassungsvoraussetzung:

Veranstaltung Prozedurale Programmierung oder gleichwertige Programmierkenntnisse in imperativer Programmierung

Empfohlene Vorkenntnisse:

Zwingende Voraussetzung ist die Beherrschung imperativer Programmierung (C, Pascal, Fortran oder ähnlich). Sie sollten also z.B. einfache Datentypen (integer, double, char, bool), arrays, if-then-else, for, while, Prozedur- bzw. Funktionsaufrufe und Zeiger kennen und in eigenen Programmen damit experimentiert haben, also auch Editor, Linker, Compiler und Debugger nutzen können. Die Veranstaltung beginnt mit der Einführung von Objekten, setzt also auf oben genannte Grundlagen auf. Dieser Hinweis ist insbesondere wichtig für Studiengänge wie AIW, GES, LUM da oben genannte Voraussetzungen dort **nicht** Bestandteil des Studienplans sind, sondern zu den Studienvoraussetzungen dieser Studiengänge zählen. Die Studiengänge ET, CI und IIW besitzen die erforderlichen Vorkenntnisse aus der Veranstaltung Prozedurale Programmierung im ersten Semester.

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Studierende können die Grundzüge des Software-Entwurfs wie den Entwurf einer Klassenarchitektur unter Einbeziehung vorhandener Klassenbibliotheken und Entwurfsmuster erklären.
Studierende können grundlegende Datenstrukturen der diskreten Mathematik beschreiben sowie wichtige Algorithmen zum Sortieren und Suchen bezüglich ihrer Komplexität bewerten.

Fertigkeiten:

Studierende sind in der Lage,

- Software mit gegebenen Entwurfsmustern, unter Verwendung von Klassenhierarchien und Polymorphie zu entwerfen.
- Softwareentwicklung und Tests unter Verwendung von Versionsverwaltungssystemen und google Test durchzuführen.
- Sortierung und Suche nach Daten effizient durchzuführen.
- die Komplexität von Algorithmen abzuschätzen.

Personale Kompetenzen:

Sozialkompetenz:

Studierende können in Teams arbeiten und in Foren kommunizieren.

Selbstständigkeit:

Studierende sind in der Lage selbständig über einen Zeitraum von 2-3 Wochen, unter Verwendung von SVN Repository und google Test, Programmieraufgaben z.B. LZW Datenkompression zu lösen.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 110, Präsenzstudium: 70

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
Computer Science: Kernqualifikation: Pflicht
Computer Science myTrack: Kernqualifikation: Pflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht
Technomathematik: Kernqualifikation: Pflicht

Lehrveranstaltung: Objektorientierte Programmierung, Algorithmen und Datenstrukturen (Vorlesung)

Dozenten:

Prof. Rolf-Rainer Grigat

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Objektorientierte Analyse und Entwurf:

- Objektorientierte Programmierung in C++ und Java
- generische Programmierung
- UML
- Entwurfsmuster

Datenstrukturen und Algorithmen:

- Komplexität von Algorithmen
- Suchen, Sortieren, Hashing,
- Stapel, Schlangen, Listen
- Bäume (AVL, Heap, 2-3-4, Trie, Huffman, Patricia, B),
- Mengen, Prioritätswarteschlangen
- gerichtete und ungerichtete Graphen (Spannbäume, kürzeste und längste Wege)

Literatur:

Skriptum

Lehrveranstaltung: Objektorientierte Programmierung, Algorithmen und Datenstrukturen (Gruppenübung)

Dozenten:

Prof. Rolf-Rainer Grigat

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Logik, Automatentheorie und Formale Sprachen	Vorlesung	2
Logik, Automatentheorie und Formale Sprachen	Gruppenübung	2

Modulverantwortlich:

Prof. Tobias Knopp

Zulassungsvoraussetzung:

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:

Wissen:

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.

Fertigkeiten:

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.

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Logic, Automata Theory and Formal Languages (Vorlesung)

Dozenten:

Prof. Tobias Knopp

Sprachen:

EN

Zeitraum:

SS

Inhalt:

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 ϵ -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 μ -calculus
23. Characterization of regular languages by monadic second-order logic (MSO)

Literatur:

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: Logic, Automata Theory and Formal Languages (Gruppenübung)

Dozenten:

Prof. Tobias Knopp

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Software Engineering

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Software-Engineering	Vorlesung	2
Software-Engineering	Gruppenübung	2

Modulverantwortlich:

Prof. Sibylle Schupp

Zulassungsvoraussetzung:

- Procedural programming or Functional programming
- Object-oriented programming, algorithms, and data structures

Empfohlene Vorkenntnisse:

- Automata theory and formal languages

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

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.

Fertigkeiten:

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.

Personale Kompetenzen:

Sozialkompetenz:

Students practice peer programming. They explain problems and solutions to their peer. They communicate in English.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht

Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht

Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Software Engineering (Vorlesung)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

SS

Inhalt:

- 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)

Modulhandbuch - Bachelor of Science "Computer Science"

- 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: Software Engineering (Gruppenübung)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Mathematical Analysis

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Mathematische Analysis	Vorlesung	4
Mathematische Analysis	Hörsaalübung	2
Mathematische Analysis	Gruppenübung	2

Modulverantwortlich:

Prof. Marko Lindner

Zulassungsvoraussetzung:

Empfohlene Vorkenntnisse:

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- Students can name the basic concepts in analysis. They are able to explain them using appropriate examples.
- Students can discuss logical connections between these concepts. They are capable of illustrating these connections with the help of examples.
- They know proof strategies and can reproduce them.

Fertigkeiten:

- Students can model problems in analysis with the help of the concepts studied in this course. Moreover, they are capable of solving them by applying established methods.
- Students are able to discover and verify further logical connections between the concepts studied in the course.
- For a given problem, the students can develop and execute a suitable approach, and are able to critically evaluate the results.

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

8 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 128, Präsenzstudium: 112

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

General Engineering Science: Kernqualifikation: Pflicht

Lehrveranstaltung: Mathematical Analysis (Vorlesung)

Dozenten:

Dr. Francisco Javier Hoecker-Escuti

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Convergence, sequences, and series

Continuity

Elementary functions

Differential calculus

Integral calculus

Sequences of functions

Literatur:

Königsberger: Analysis

Forster: Analysis

Lehrveranstaltung: Mathematical Analysis (Hörsaalübung)

Dozenten:

Dr. Francisco Javier Hoecker-Escuti

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Mathematical Analysis (Gruppenübung)

Dozenten:

Dr. Francisco Javier Hoecker-Escuti

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Grundlagen der Betriebswirtschaftslehre	Vorlesung	4
Projekt Entrepreneurship	Problemorientierte Lehrveranstaltung	2

Modulverantwortlich:

Prof. Christoph Ihl

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Schulkenntnisse in Mathematik und Wirtschaft

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden können...

- grundlegende Begriffe und Kategorien aus dem Bereich Wirtschaft und Management benennen und erklären
- grundlegende Aspekte wettbewerblichen 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)

Fertigkeiten:

Die Studierenden können

- 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:

Sozialkompetenz:

Die Studierenden sind in der Lage

- 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

Selbstständigkeit:

Die Studierenden sind in der Lage

- Ein Projekt in einem Team zu bearbeiten und einer Lösung zuzuführen
- unter Anleitung einen Projektbericht zu verfassen

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 96, Präsenzstudium: 84

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Elektrotechnik: Pflicht

Modulhandbuch - Bachelor of Science "Computer Science"

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Verfahrenstechnik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Bioverfahrenstechnik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Energie- und Umwelttechnik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Bau- und Umweltingenieurwesen: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Mediziningenieurwesen: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Schiffbau: Pflicht
Bau- und Umweltingenieurwesen: Kernqualifikation: Pflicht
Bioverfahrenstechnik: Kernqualifikation: Pflicht
Computer Science: Kernqualifikation: Pflicht
Computer Science myTrack: Kernqualifikation: Pflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
Energie- und Umwelttechnik: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Bau- und Umweltingenieurwesen: Pflicht
General Engineering Science: Vertiefung Bioverfahrenstechnik: Pflicht
General Engineering Science: Vertiefung Elektrotechnik: Pflicht
General Engineering Science: Vertiefung Energie- und Umwelttechnik: Pflicht
General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
General Engineering Science: Vertiefung Maschinenbau: Pflicht
General Engineering Science: Vertiefung Mediziningenieurwesen: Pflicht
General Engineering Science: Vertiefung Schiffbau: Pflicht
General Engineering Science: Vertiefung Verfahrenstechnik: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
Logistik und Mobilität: Kernqualifikation: Pflicht
Maschinenbau: Kernqualifikation: Pflicht
Mechatronik: Kernqualifikation: Pflicht
Schiffbau: Kernqualifikation: Pflicht
Technomathematik: Kernqualifikation: Pflicht
Verfahrenstechnik: Kernqualifikation: Pflicht

Lehrveranstaltung: Grundlagen der Betriebswirtschaftslehre (Vorlesung)

Dozenten:

Prof. Christoph Ihl, Prof. Thorsten Blecker, Prof. Christian Lühje, Prof. Christian Ringle, Prof. Kathrin Fischer, Prof. Cornelius Herstatt, Prof. Wolfgang Kersten, Prof. Matthias Meyer, Prof. Thomas Wrona

Sprachen:

DE

Zeitraum:

WS/SS

Inhalt:

- 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

Literatur:

Bamberg, G., Coenenberg, A.: Betriebswirtschaftliche Entscheidungslehre, 14. Aufl., München 2008
Eisenführ, F., Weber, M.: Rationales Entscheiden, 4. Aufl., Berlin et al. 2003
Heinhold, M.: Buchführung in Fallbeispielen, 10. Aufl., Stuttgart 2006.
Kruschwitz, L.: Finanzmathematik. 3. Auflage, München 2001.

Modulhandbuch - Bachelor of Science "Computer Science"

Pellens, B., Fülber, R. U., Gassen, J., Sellhorn, T.: Internationale Rechnungslegung, 7. Aufl., Stuttgart 2008.

Schweitzer, M.: Planung und Steuerung, in: Bea/Friedl/Schweitzer: Allgemeine Betriebswirtschaftslehre, Bd. 2: Führung, 9. Aufl., Stuttgart 2005.

Weber, J., Schäffer, U. : Einführung in das Controlling, 12. Auflage, Stuttgart 2008.

Weber, J./Weißberger, B.: Einführung in das Rechnungswesen, 7. Auflage, Stuttgart 2006.

Lehrveranstaltung: Projekt Entrepreneurship (Problemorientierte Lehrveranstaltung)

Dozenten:

Prof. Christoph Ihl

Sprachen:

DE

Zeitraum:

WS/SS

Inhalt:

Inhalt ist die eigenständige Erarbeitung eines Gründungsprojekts, von der ersten Idee bis zur fertigen Konzeption, wobei die betriebswirtschaftlichen Grundkenntnisse aus der Vorlesung "Grundlagen der Betriebswirtschaftslehre" zum Einsatz kommen sollen. Die Erarbeitung erfolgt in Teams und unter Anleitung eines Mentors.

Literatur:

Relevante Literatur aus der korrespondierenden Vorlesung.

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Technische Informatik	Vorlesung	3
Technische Informatik	Gruppenübung	1

Modulverantwortlich:

Prof. Heiko Falk

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Grundkenntnisse der Elektrotechnik

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Dieses Modul vermittelt Grundkenntnisse der Funktionsweise von Rechensystemen. Abgedeckt werden die Ebenen von der Assemblerprogrammierung bis zur Gatterebene. Das Modul behandelt folgende Inhalte:

- Einführung
- Kombinatorische Logik: Gatter, Boolesche Algebra, Schaltfunktionen, Synthese von Schaltungen, Schaltnetze
- Sequentielle Logik: Flip-Flops, Schaltwerke, systematischer Schaltwerkentwurf
- 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

Fertigkeiten:

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.

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.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Kernqualifikation: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 Elektrotechnik: Kernqualifikation: Pflicht
 Elektrotechnik myTrack: Kernqualifikation: Pflicht
 General Engineering Science: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Mechatronik: Kernqualifikation: Pflicht
 Technomathematik: Vertiefung Informatik: Wahlpflicht

Dozenten:

Prof. Heiko Falk

Sprachen:

DE

Zeitraum:

WS

Inhalt:

- Einführung
- Kombinatorische Logik
- Sequentielle Logik
- Technologische Grundlagen
- Zahlendarstellungen und Rechnerarithmetik
- Grundlagen der Rechnerarchitektur
- Speicher-Hardware
- Ein-/Ausgabe

Literatur:

- 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: Technische Informatik (Gruppenübung)

Dozenten:

Prof. Heiko Falk

Sprachen:

DE

Zeitraum:

WS

Inhalt:

1. Einführung

- Grundlagen der Digitaltechnik
- Analog versus Digital
- Gatter und Flipflops
- Aspekte der Digitaltechnik
- Integrierte Schaltkreise
- Digitale Systeme
- Time-to-Market

2. Zahlensysteme und Codierung

- Zahlensysteme
- Rechnerinterne Zahlenformate
- Arithmetische Operationen im Dualsystem
- Zahlen- und Zeichencodes
- Fehlererkennende und -korrigierende Codes
- Codes zur seriellen Datenübertragung
- Binäre Vorsätze für Zweierpotenzen

3. Digitale Schaltungstechnik

- Logische Signale und Gatter
- Logikfamilien
- CMOS-Logik
- CMOS-Schaltungstechnik: Elektrisches Verhalten
- CMOS-Schaltungen für Ein- und Ausgänge
- Bipolare Logik und TTL-Schaltungstechnik
- CMOS-Logikfamilien
- CMOS/TTL-Schnittstelle

4. Schaltnetze (Grundlagen)

- Boolesche Algebra
- Analyse kombinatorischer Schaltungen
- Synthese kombinatorischer Schaltungen

- Minimierungsverfahren
- Störimpulse bei digitalen Schaltungen

5. Schaltnetze (Anwendungen)

- Standards zur Dokumentation
- Zeitverhalten digitaler Schaltungen
- Decodierer und Codierer
- Tri-State-Logikgatter und Busse
- Multiplexer und Demultiplexer
- Präfix-Logik und Paritätsschaltungen
- Komparatoren
- Addierer und Subtrahierer
- Multiplizierer
- Barrel Shifter
- Arithmetisch-Logische Einheit (ALU)

6. Schaltwerke (Grundlagen)

- Zustandsbegriff und Taktsignal
- Bistabile Speicherelemente
- Asynchrone Speicherelemente
- Synchrone taktzustandsgesteuerte Speicherelemente
- Synchrone taktf flankengesteuerte Speicherelemente
- Übersicht: Latches und Flipflops
- Analyse von Schaltwerken
- Klassisches Design von Schaltwerken
- Design von Schaltwerken mit Zustandsübergangsgraphen
- Design von Schaltwerken mit VHDL
- Hierarchische Schaltwerkstrukturen

7. Schaltwerke (Anwendungen)

- Standards zur Dokumentation
- Latches und Flipflops
- Zähler
- Schieberegister
- Iterative Schaltnetze versus Schaltwerke
- Design-Methodik für synchrone Systeme
- Problematik bei synchronen Designs

8. Speicher, PLDs, CPLDs und FPGAs

- ROM, SRAM, DRAM, SDRAM
- Programmable Logic Devices (PLDs)
- Complex Programmable Logic Devices (CPLDs)
- Field-Programmable Gate Arrays (FPGAs)

9. Mikroprozessortechnik (Grundlagen)

- Historisches
- Von-Neumann-Architektur
- Komponenten eines Mikroprozessorsystems

Literatur:

- S. Voigt, *Skript zur Vorlesung „Technische Informatik“*
- J. Wakerly, *Digital Design: Principles and Practices*, 4. Auflage, 2010, Pearson Prentice Hall, ISBN: 978-0-13-613987-4
- D. Hoffmann, *Grundlagen der Technischen Informatik*, 2. Auflage, 2010, Carl Hanser Verlag, ISBN: 978-3-446-42150-9

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Rechnernetze und Internet-Sicherheit	Vorlesung	3
Rechnernetze und Internet-Sicherheit	Gruppenübung	1

Modulverantwortlich:

Prof. Andreas Timm-Giel

Zulassungsvoraussetzung:

Empfohlene Vorkenntnisse:

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

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.

Fertigkeiten:

Students are able to analyse common Internet protocols and evaluate the use of them in different domains.

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Students can select relevant parts out of high amount of professional knowledge and can independently learn and understand it.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 Elektrotechnik: Kernqualifikation: Wahlpflicht
 Elektrotechnik myTrack: Kernqualifikation: Wahlpflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Computer Networks and Internet Security (Vorlesung)

Dozenten:

Prof. Andreas Timm-Giel, Prof. Dieter Gollmann

Sprachen:

EN

Zeitraum:

WS

Inhalt:

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.

In the second part of the lecture an introduction to Internet security is given.

This class comprises:

- 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:

- 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

Further literature is announced at the beginning of the lecture.

Lehrveranstaltung: Computer Networks and Internet Security (Gruppenübung)

Dozenten:

Prof. Andreas Timm-Giel, Prof. Dieter Gollmann

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Mathematik III

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Analysis III	Vorlesung	2
Analysis III	Gruppenübung	1
Analysis III	Hörsaalübung	1
Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen)	Vorlesung	2
Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen)	Gruppenübung	1
Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen)	Hörsaalübung	1

Modulverantwortlich:

Prof. Anusch Taraz

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Mathematik I + II

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- Studierende können die grundlegenden Begriffe aus dem Gebiet der Analysis und Differentialgleichungen 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.

Fertigkeiten:

- Studierende können Aufgabenstellungen aus dem Gebiet der Analysis und Differentialgleichungen mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.
- Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.
- Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.

Personale Kompetenzen:

Sozialkompetenz:

- 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.

Selbstständigkeit:

- 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.

Leistungspunkte:

8 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 128, Präsenzstudium: 112

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Kernqualifikation: Pflicht
 Bioverfahrenstechnik: Kernqualifikation: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 Elektrotechnik: Kernqualifikation: Pflicht
 Elektrotechnik myTrack: Kernqualifikation: Pflicht
 Energie- und Umwelttechnik: Kernqualifikation: Pflicht
 General Engineering Science: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht

Modulhandbuch - Bachelor of Science "Computer Science"

Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
Maschinenbau: Kernqualifikation: Pflicht
Mechatronik: Kernqualifikation: Pflicht
Schiffbau: Kernqualifikation: Pflicht
Verfahrenstechnik: Kernqualifikation: Pflicht

Lehrveranstaltung: Analysis III (Vorlesung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Grundzüge der Differential- und Integralrechnung mehrerer Variablen:

- Differentialrechnung mehrerer Veränderlichen
- Mittelwertsätze und Taylorscher Satz
- Extremwertbestimmung
- Implizit definierte Funktionen
- Extremwertbestimmung bei Gleichungsnebenbedingungen
- Newton-Verfahren für mehrere Variablen
- Bereichsintegrale
- Kurven- und Flächenintegrale
- Integralsätze von Gauß und Stokes

Literatur:

- R. Ansorge, H. J. Oberle: Mathematik für Ingenieure, Band 2; Verlag Wiley-VCH, Berlin, Weinheim, New York, 2000
 - H.J. Oberle, K. Rothe, Th. Sonar: Mathematik für Ingenieure, Band 3: Aufgaben und Lösungen; Verlag Wiley-VCH, Berlin, Weinheim, New York, 2000.
-

Lehrveranstaltung: Analysis III (Gruppenübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Analysis III (Hörsaalübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (Vorlesung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Grundzüge der Theorie und Numerik gewöhnlicher Differentialgleichungen

- Einführung und elementare Methoden
- Existenz und Eindeutigkeit bei Anfangswertaufgaben
- Lineare Differentialgleichungen
- Stabilität und qualitatives Lösungsverhalten
- Randwertaufgaben und Grundbegriffe der Variationsrechnung
- Eigenwertaufgaben
- Numerische Verfahren zur Integration von Anfangs- und Randwertaufgaben
- Grundtypen bei partiellen Differentialgleichungen

Literatur:

- R. Ansorge, H. J. Oberle: Mathematik für Ingenieure, Band 2; Verlag Wiley-VCH, Berlin, Weinheim, New York, 2000
- H.J. Oberle, K. Rothe, Th. Sonar: Mathematik für Ingenieure, Band 3: Aufgaben und Lösungen; Verlag Wiley-VCH, Berlin, Weinheim, New York, 2000.

Lehrveranstaltung: Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (Gruppenübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Differentialgleichungen 1 (Gewöhnliche Differentialgleichungen) (Hörsaalübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Introduction to Information Security

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Einführung in die Informationssicherheit	Vorlesung	3
Einführung in die Informationssicherheit	Gruppenübung	2

Modulverantwortlich:

Prof. Dieter Gollmann

Zulassungsvoraussetzung:

None

Empfohlene Vorkenntnisse:

Basics of Computer Science

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Students can

- 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.

Fertigkeiten:

Students can

- 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:

Sozialkompetenz:

Students are capable of appreciating the impact of security problems on those affected and of the potential responsibilities for their resolution.

Selbstständigkeit:

None

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 110, Präsenzstudium: 70

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht

Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht

International Production Management: Technischer Ergänzungskurs Kernfächer: Wahlpflicht

Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Introduction to Information Security (Vorlesung)

Dozenten:

Prof. Dieter Gollmann

Sprachen:

EN

Zeitraum:

WS

Inhalt:

- Fundamental concepts
- Passwords & biometrics
- Introduction to cryptography
- Sessions, SSL/TLS

Modulhandbuch - Bachelor of Science "Computer Science"

- 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: Introduction to Information Security (Gruppenübung)

Dozenten:

Prof. Dieter Gollmann

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Berechenbarkeit und Komplexität

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Berechenbarkeit und Komplexität	Vorlesung	2
Berechenbarkeit und Komplexität	Gruppenübung	2

Modulverantwortlich:

Prof. Karl-Heinz Zimmermann

Zulassungsvoraussetzung:

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:

Wissen:

Wissen: Die Studierenden kennen

- 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;
- das Hilberts zehnte Problem;
- die Komplexitätsklassen P und NP und deren Unterscheidung;
- das Konzept der NP-Vollständigkeit sowie den Satz von Cook.

Fertigkeiten:

Fertigkeiten: Die Studierenden können

- 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.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachbüchern und anderweitiger Literatur selbständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.

Leistungspunkte:

6 LP

Prüfung:

Mündliche Prüfung

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Berechenbarkeit und Komplexität (Vorlesung)

Dozenten:

Prof. Karl-Heinz Zimmermann

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Literatur:

Lehrveranstaltung: Berechenbarkeit und Komplexität (Gruppenübung)

Dozenten:

Prof. Karl-Heinz Zimmermann

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Literatur:

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Signale und Systeme	Vorlesung	3
Signale und Systeme	Hörsaalübung	1

Modulverantwortlich:

Prof. Gerhard Bauch

Zulassungsvoraussetzung:

Mathematik 1-3

Empfohlene Vorkenntnisse:

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:

Wissen:

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.

Fertigkeiten:

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.

Personale Kompetenzen:

Sozialkompetenz:

Die Studierenden können fachspezifische Aufgaben gemeinsam bearbeiten.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Elektrotechnik: Pflicht
 Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Allgemeine Ingenieurwissenschaften: Vertiefung Verfahrenstechnik: Pflicht
 Allgemeine Ingenieurwissenschaften: Vertiefung Bioverfahrenstechnik: Pflicht
 Allgemeine Ingenieurwissenschaften: Vertiefung Bau- und Umweltingenieurwesen: Pflicht
 Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau: Pflicht
 Allgemeine Ingenieurwissenschaften: Vertiefung Medizingenieurwesen: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 Elektrotechnik: Kernqualifikation: Pflicht
 Elektrotechnik myTrack: Kernqualifikation: Pflicht
 General Engineering Science: Vertiefung Bau- und Umweltingenieurwesen: Pflicht
 General Engineering Science: Vertiefung Bioverfahrenstechnik: Pflicht
 General Engineering Science: Vertiefung Elektrotechnik: Pflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 General Engineering Science: Vertiefung Maschinenbau: Pflicht
 General Engineering Science: Vertiefung Medizingenieurwesen: Pflicht
 General Engineering Science: Vertiefung Verfahrenstechnik: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Mechatronik: Kernqualifikation: Pflicht

Lehrveranstaltung: Signale und Systeme (Vorlesung)

Dozenten:

Prof. Gerhard Bauch

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

- Elementare Klassifizierung und Beschreibung zeitkontinuierlicher und zeitdiskreter deterministischer Signale und Systemen
- Faltung
- Leistung und Energie von Signalen
- Korrelationsfunktionen deterministischer Signale
- Lineare zeitinvariante (LTI) Systeme
- Signaltransformationen:
 - Fourier-Reihe
 - Fourier Transformation
 - Laplace Transformation
 - Zeitdiskrete Fouriertransformation
 - Diskrete Fouriertransformation (DFT), Fast Fourier Transform (FFT)
 - Z-Transformation
- Analyse und Entwurf von LTI-Systemen in Zeit- und Frequenzbereich
- Grundlegende Filtertypen
- Abtastung, Abtasttheorem
- Grundlagen rekursiver und nicht-rekursiver zeitdiskreter Filter

Literatur:

- T. Frey , M. Bossert , Signal- und Systemtheorie, B.G. Teubner Verlag 2004
 - K. Kammeyer, K. Kroschel, Digitale Signalverarbeitung, Teubner Verlag.
 - 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: Signale und Systeme (Hörsaalübung)

Dozenten:

Prof. Gerhard Bauch

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Stochastics

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Stochastik	Vorlesung	2
Stochastik	Gruppenübung	2

Modulverantwortlich:

NN

Zulassungsvoraussetzung:

none

Empfohlene Vorkenntnisse:

- Calculus
- Discrete algebraic structures (combinatorics)
- Propositional logic

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Students can explain the main definitions of probability, and they can give basic definitions of modeling elements (random variables, events, dependence, independence assumptions) used in discrete and continuous settings (joint and marginal distributions, density functions). Students can describe characteristic notions such as expected values, variance, standard deviation, and moments. Students can define decision problems and explain algorithms for solving these problems (based on the chain rule or Bayesian networks). Algorithms, or estimators as they are called, can be analyzed in terms of notions such as bias of an estimator, etc. Student can describe the main ideas of stochastic processes and explain algorithms for solving decision and computation problem for stochastic processes. Students can also explain basic statistical detection and estimation techniques.

Fertigkeiten:

Students can apply algorithms for solving decision problems, and they can justify whether approximation techniques are good enough in various application contexts, i.e., students can derive estimators and judge whether they are applicable or reliable.

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht

Lehrveranstaltung: Stochastics (Vorlesung)

Dozenten:

NN

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Foundations of probability theory

- Definitions of probability, conditional probability
- Random variables, dependencies, independence assumptions,
- Marginal and joint probabilities
- Distributions and density functions
- Characteristics: expected values, variance, standard deviation, moments

Practical representations for joint probabilities

- Bayessche Netzwerke
- Semantik, Entscheidungsprobleme, exakte und approximative Algorithmen

Stochastic processes

- Stationarity, ergodicity
- Correlations
- Dynamic Bayesian networks, Hidden Markov networks, Kalman filters, queues

Detection & estimation

- Detectors
- Estimation rules and procedures
- Hypothesis and distribution tests
- Stochastic regression

Literatur:

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: Stochastics (Gruppenübung)

Dozenten:

NN

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Graphentheorie und Optimierung	Vorlesung	2
Graphentheorie und Optimierung	Gruppenübung	2

Modulverantwortlich:

Prof. Anusch Taraz

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

- Diskrete Algebraische Strukturen
- Mathematik I

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- 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.

Fertigkeiten:

- 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.

Personale Kompetenzen:

Sozialkompetenz:

- 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.

Selbstständigkeit:

- 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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
Computer Science: Kernqualifikation: Pflicht
Computer Science myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht
Technomathematik: Vertiefung Mathematik: Wahlpflicht

Lehrveranstaltung: Graphentheorie und Optimierung (Vorlesung)

Dozenten:

Prof. Anusch Taraz

Sprachen:

DE

Zeitraum:

SS

Inhalt:

- 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:

- M. Aigner: Diskrete Mathematik, Vieweg, 2004
 - 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: Graphentheorie und Optimierung (Gruppenübung)

Dozenten:

Prof. Anusch Taraz

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Betriebssysteme

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Betriebssysteme	Vorlesung	2
Betriebssysteme	Gruppenübung	2

Modulverantwortlich:

Prof. Volker Turau

Zulassungsvoraussetzung:

- Prozedurales Programmieren
- Objekt-orientierte Programmierung, Algorithmen und Datenstrukturen

Empfohlene Vorkenntnisse:

- 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:

Wissen:

Studierende können die wichtigsten Abstraktion von Betriebssystem erklären (Prozess, virtueller Speicher, Datei, Deadlock, Lifelock). 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.

Fertigkeiten:

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.

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Computer Science: Kernqualifikation: Pflicht
 Computer Science myTrack: Kernqualifikation: Pflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
 Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
 Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Betriebssysteme (Vorlesung)

Dozenten:

Prof. Volker Turau

Sprachen:

DE

Zeitraum:

SS

Inhalt:

- 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: Betriebssysteme (Gruppenübung)

Dozenten:

Prof. Volker Turau

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Proseminar Computerorientierte Mathematik	Seminar	2
Proseminar Informatik	Seminar	2
Seminar	Seminar	2

Modulverantwortlich:

Dozenten des SD E

Zulassungsvoraussetzung:

Empfohlene Vorkenntnisse:

Grundkenntnisse in Informatik

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Fertigkeiten:

Personale Kompetenzen:

Sozialkompetenz:

- Studierende verstehen, dass Methoden der Informatik und Mathematik anwendungsübergreifend entwickelt werden und dass eine wesentliche Leistung des Informatik-Ingenieurs zum Einen in der fachgerechten Anwendung der Methoden liegt und zum Anderen darin besteht, andere (Auftraggeber, Projektpartner, Kollegen, ...) überzeugen zu können, dass eine Methode sich (in einem gewissen Sinne) optimal eignet.
- Studierenden erläutern die in einem wissenschaftlichen Aufsatz geschilderten Probleme und die im Aufsatz entwickelten Lösungen in einem Fachgebiet der Informatik oder Mathematik, bewerten die vorgeschlagenen Lösungen in einem Vortrag und reagieren auf wissenschaftliche Nachfragen, Ergänzungen und Kommentare
- Studierenden beschreiben wissenschaftliche Fragestellungen in einem Fachgebiet der Informatik, des Ingenieurwesens oder der Mathematik und erläutern in einem Vortrag einen von ihnen entwickelten Ansatz zu dessen Lösung und reagieren dabei angemessen auf Nachfragen, Ergänzungen und Kommentare.

Selbstständigkeit:

- Die Studierenden bewerten selbständig Vor- und Nachteile von Repräsentationsformalismen für bestimmte Aufgaben, vergleichen verschiedene Algorithmen und Datenstrukturen sowie Programmiersprachen und Programmierwerkzeuge, und sie wählen eigenverantwortlich die jeweils beste Lösung aus.
- Die AbsolventInnen erarbeiten sich selbständig ein kleines, sehr klar abgegrenztes wissenschaftliches Teilgebiet, können dieses in einer Präsentation vorstellen und verfolgen aktiv die Präsentationen anderer Studierender, so dass ein interaktiver Diskurs über ein wissenschaftliches Thema entsteht.

Leistungspunkte:

6 LP

Arbeitsaufwand in Stunden:

Eigenstudium: 96, Präsenzstudium: 84

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
Computer Science: Kernqualifikation: Pflicht
Computer Science myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht

Lehrveranstaltung: Proseminar Computerorientierte Mathematik (Seminar)

Dozenten:

Dozenten des SD E

Sprachen:

DE/EN

Zeitraum:

WS

Prüfungsform:

Referat

Benotet:

Nein

Leistungspunkte:

2

Inhalt:

- Seminarvorträge der teilnehmenden Studierenden. Seminarthemen aus dem Bereich Computerorientierte Mathematik werden vom Veranstalter bekanntgegeben
- Aktive Teilnahme an der Diskussion.

Literatur:

Wird vom Seminarveranstalter bekanntgegeben.

Lehrveranstaltung: Proseminar Informatik (Seminar)

Dozenten:

Dozenten des SD E

Sprachen:

DE/EN

Zeitraum:

WS

Prüfungsform:

Referat

Benotet:

Nein

Leistungspunkte:

2

Inhalt:

- Seminarvorträge der teilnehmenden Studierenden. Seminarthemen aus dem Bereich Informatik werden vom Veranstalter bekanntgegeben
- Aktive Teilnahme an der Diskussion.

Literatur:

Wird vom Seminarveranstalter bekanntgegeben.

Lehrveranstaltung: Seminar (Seminar)

Dozenten:

Dozenten des SD E

Sprachen:

DE/EN

Zeitraum:

WS

Prüfungsform:

Referat

Benotet:

Nein

Leistungspunkte:

2

Inhalt:

- Seminarvorträge der teilnehmenden Studierenden. Seminarthemen aus dem Bereich Computerorientierte Mathematik oder Informatik werden vom Veranstalter bekanntgegeben
- Aktive Teilnahme an der Diskussion.

Literatur:

Wird vom Seminarveranstalter bekanntgegeben.

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Software-Fachpraktikum	Laborpraktikum	6

Modulverantwortlich:

Dozenten des SD E

Zulassungsvoraussetzung:

keine

Empfohlene Vorkenntnisse:

Grundlagen der Informatik (Semester 1 bis 4)

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden kennen wichtige Aspekte und typische Phasen der Software-Projektentwicklung (wie Planung, Spezifikation, Implementierung, Validierung, Wartung oder Dokumentation), die in der Praxis mehr oder weniger ausgeprägt zum Tragen kommen.

Fertigkeiten:

Die Studierenden können

- typische Stufen der Software-Projektentwicklung beschreiben,
- eng umrissene Teilaufgaben in einem Software-Projekt innerhalb einer Gruppe oder eigenständig bearbeiten.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Schriftliche Ausarbeitung

Arbeitsaufwand in Stunden:

Eigenstudium: 96, Präsenzstudium: 84

Zuordnung zu folgenden Curricula:

Computer Science: Kernqualifikation: Pflicht

Computer Science myTrack: Kernqualifikation: Pflicht

Lehrveranstaltung: Software-Fachpraktikum (Laborpraktikum)

Dozenten:

Dozenten des SD E

Sprachen:

DE/EN

Zeitraum:

WS/SS

Inhalt:

Literatur:

Fachmodule der Vertiefung Computational Mathematics

Modul: Rechnergestützte Geometrie

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Rechnergestützte Geometrie	Vorlesung	2
Rechnergestützte Geometrie	Gruppenübung	2

Modulverantwortlich:

Dr. Prashant Batra

Zulassungsvoraussetzung:

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:

Wissen:

Studierende können die grundlegenden Begriffe der Rechnergestützten Geometrie benennen, mathematisch exakt beschreiben und anhand von Beispielen erklären.

Studierende sind in der Lage, logische Zusammenhänge zwischen diesen Konzepten zu diskutieren und anhand von Beispielen zu erläutern.

Fertigkeiten:

Studierende können Aufgabenstellungen aus der Rechnergestützten Geometrie mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
 Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
 Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
 Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Rechnergestützte Geometrie (Vorlesung)

Dozenten:

Dr. Prashant Batra

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Konstruktion einer konvexen Hülle von n Punkten
Triangulierung eines schlichten Polygons
Konstruktion einer Delaunay-Triangulation, eines Voronoi-Diagramms
Algorithmen und Datenstrukturen zum Bestimmen eines Arrangements, eines Ham-Sandwich-Cuts.
des Schnitts von Halbebenen, der Optimierung eines linearen Funktionals.
Effiziente Bestimmung aller Schnittpunkte von (orthogonalen) Streckensegmenten
Approximative Berechnung des Durchmessers einer Punktemenge
Inkrementelle randomisierte Algorithmen
Grundlagen der Gitterpunktlehre, LLL-Algorithmus und Anwendungen in der ganzzahligen Optimierung

Grundlagen der Bewegungsplanung

Literatur:

Computational Geometry Algorithms and Applications Authors:

- Prof. Dr. Mark de Berg,
- Dr. Otfried Cheong,
- Dr. Marc van Kreveld,
- Prof. Dr. Mark Overmars

Springer e-Book: <http://dx.doi.org/10.1007/978-3-540-77974-2>

Algorithmische Geometrie : Grundlagen, Methoden, Anwendungen / Rolf Klein

Verfasser: Klein, Rolf

Ausgabe: 2., vollst. überarb. Aufl.

Erschienen: Berlin [u.a.] : Springer, 2005

Umfang: XI, 392 S. : graph. Darst.

Springer e-Book: <http://dx.doi.org/10.1007/3-540-27619-X>

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

Computational geometry : an introduction / Franco P. Preparata; Michael Ian Shamos

Verfasser: Preparata, Franco P. ; Shamos, Michael Ian

Ausgabe: Corr. and expanded 2. printing.

Erschienen: New York [u.a.] : Springer, 1988

Umfang: XIV, 398 S. : graph. Darst.

Schriftenreihe: Texts and monographs in computer science

ISBN: 3-540-96131-3

0-387-96131-3

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)

Lehrveranstaltung: Rechnergestützte Geometrie (Gruppenübung)

Dozenten:

Dr. Prashant Batra

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Numerische Mathematik I

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Numerische Mathematik I	Vorlesung	2
Numerische Mathematik I	Gruppenübung	2

Modulverantwortlich:

Prof. Sabine Le Borne

Zulassungsvoraussetzung:

- Mathematik I + II für Ingenieurstudierende (deutsch oder englisch)

oder

- Analysis & Lineare Algebra I + II für Technomathematiker

Empfohlene Vorkenntnisse:

- Vorlesungsinhalte der Veranstaltungen der Zulassungsvoraussetzungen
- MATLAB Grundkenntnisse

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Studierende können

- numerische Verfahren zur Interpolation, Integration, Lösung von Ausgleichproblemen, Lösung von Eigenwertproblemen und nichtlinearen Nullstellenproblemen benennen und deren Kernideen erläutern,
- Konvergenzaussagen zu den numerischen Methoden wiedergeben,
- Aspekte der praktischen Durchführung numerischer Verfahren im Hinblick auf Rechenzeit und Speicherbedarf erklären.

Fertigkeiten:

Studierende sind in der Lage,

- numerische Methoden in MATLAB zu implementieren, anzuwenden und zu vergleichen,
- das Konvergenzverhalten numerischen Methoden in Abhängigkeit vom gestellten Problem und des verwendeten Lösungsalgorithmus zu begründen,
- zu gegebener Problemstellung einen geeigneten Lösungsansatz auszuwählen und durchzuführen.

Personale Kompetenzen:

Sozialkompetenz:

Studierende können

- in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten, sich theoretische Grundlagen erklären sowie bei praktischen Implementierungsaspekten der Algorithmen unterstützen.

Selbstständigkeit:

Studierende sind fähig,

- selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen,
- ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Medizingenieurwesen: Pflicht
Bioverfahrenstechnik: Vertiefung A - Allgemeine Bioverfahrenstechnik: Wahlpflicht
Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
Elektrotechnik: Kernqualifikation: Wahlpflicht
Elektrotechnik myTrack: Kernqualifikation: Wahlpflicht

Modulhandbuch - Bachelor of Science "Computer Science"

General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht

General Engineering Science: Vertiefung Mediziningenieurwesen: Pflicht

General Engineering Science: Vertiefung Maschinenbau, Schwerpunkt Biomechanik: Pflicht

General Engineering Science: Vertiefung Maschinenbau, Schwerpunkt Materialien in den Ingenieurwissenschaften: Pflicht

Informatik-Ingenieurwesen: Kernqualifikation: Pflicht

Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht

Verfahrenstechnik: Vertiefung Allgemeine Verfahrenstechnik: Wahlpflicht

Lehrveranstaltung: Numerische Mathematik I (Vorlesung)

Dozenten:

Prof. Sabine Le Borne

Sprachen:

DE

Zeitraum:

WS

Inhalt:

1. Fehleranalyse: Zahldarstellung, Fehlertypen, Kondition, Stabilität
2. Interpolation: Polynom- und Splineinterpolation
3. Numerische Integration und Differentiation: Fehlerordnung, Newton-Cotes Formeln, Fehlerabschätzung, Gauss-Quadratur, adaptive Quadratur, Differenzenformel
4. Lineare Systeme: LR und Cholesky Zerlegung, Matrixnormen, Kondition
5. Lineare Ausgleichsprobleme: Normalgleichungen, Gram-Schmidt und Householder Orthogonalisierung, Singulärwertzerlegung, Regularisierung
6. Eigenwertaufgaben: Potenzmethode, inverse Iteration, QR-Algorithmus
7. Nichtlineare Gleichungssysteme: Fixpunkiteration, Nullstellenverfahren für reellwertige Funktionen, Newton und Quasi-Newton Verfahren für Systeme

Literatur:

- Stoer/Bulirsch: Numerische Mathematik 1, Springer
 - Dahmen, Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer
-

Lehrveranstaltung: Numerische Mathematik I (Gruppenübung)

Dozenten:

Prof. Sabine Le Borne

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Grundlagen der Regelungstechnik	Vorlesung	2
Grundlagen der Regelungstechnik	Gruppenübung	2

Modulverantwortlich:

Prof. Herbert Werner

Zulassungsvoraussetzung:

keine

Empfohlene Vorkenntnisse:

Grundkenntnisse der Behandlung von Signalen und Systemen im Zeit- und Frequenzbereich und der Laplace-Transformation.

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- Studierende können das Verhalten dynamischer Systeme in Zeit- und Frequenzbereich darstellen und interpretieren, und insbesondere die Eigenschaften Systeme 1. und 2. Ordnung erläutern.
- Sie können die Dynamik einfacher Regelkreise erklären und anhand von Frequenzgang und Wurzelortskurve interpretieren.
- Sie können das Nyquist-Stabilitätskriterium sowie die daraus abgeleiteten Stabilitätsreserven erklären.
- Sie können erklären, welche Rolle die Phasenreserve in der Analyse und Synthese von Regelkreisen spielt.
- Sie können die Wirkungsweise eines PID-Reglers anhand des Frequenzgangs interpretieren.
- Sie können erklären, welche Aspekte bei der digitalen Implementierung zeitkontinuierlich entworfener Regelkreise berücksichtigt werden müssen.

Fertigkeiten:

- Studierende können Modelle linearer dynamischer Systeme vom Zeitbereich in den Frequenzbereich transformieren und umgekehrt.
- Sie können das Verhalten von Systemen und Regelkreisen simulieren und bewerten.
- Sie können PID-Regler mithilfe heuristischer Einstellregeln (Ziegler-Nichols) entwerfen.
- Sie können anhand von Wurzelortskurve und Frequenzgang einfache Regelkreise entwerfen und analysieren.
- Sie können zeitkontinuierliche Modelle dynamischer Regler für die digitale Implementierung zeitdiskret approximieren.
- Sie beherrschen die einschlägigen Software-Werkzeuge (Matlab Control Toolbox, Simulink) für die Durchführung all dieser Aufgaben.

Personale Kompetenzen:

Sozialkompetenz:

Studierende können in kleinen Gruppen fachspezifische Fragen gemeinsam bearbeiten und ihre Reglerentwürfe experimentell testen und bewerten

Selbstständigkeit:

Studierende können sich Informationen aus bereit gestellten Quellen (Skript, Software-Dokumentation, Versuchsunterlagen) beschaffen und für die Lösung gegebener Probleme verwenden.

Sie können ihren Wissensstand mit Hilfe wöchentlicher On-Line Tests kontinuierlich überprüfen und auf dieser Basis ihre Lernprozesse steuern

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Kernqualifikation: Pflicht
Bioverfahrenstechnik: Kernqualifikation: Pflicht
Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
Energie- und Umwelttechnik: Kernqualifikation: Pflicht
General Engineering Science: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
Logistik und Mobilität: Vertiefung Ingenieurwissenschaft: Wahlpflicht
Maschinenbau: Kernqualifikation: Pflicht
Mechatronik: Kernqualifikation: Pflicht
Technomathematik: Vertiefung Ingenieurwissenschaften: Wahlpflicht

Lehrveranstaltung: Grundlagen der Regelungstechnik (Vorlesung)

Dozenten:

Prof. Herbert Werner

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Signale und Systeme

- Lineare Systeme, Differentialgleichungen und Übertragungsfunktionen
- Systeme 1. und 2. Ordnung, Pole und Nullstellen, Impulsantwort und Sprungantwort
- Stabilität

Regelkreise

- Prinzip der Rückkopplung: Steuerung oder Regelung
- Folgeregelung und Störunterdrückung
- Arten der Rückführung, PID-Regelung
- System-Typ und bleibende Regelabweichung
- Inneres-Modell-Prinzip

Wurzelortskurven

- Konstruktion und Interpretation von Wurzelortskurven
- Wurzelortskurven von PID-Regelkreisen

Frequenzgang-Verfahren

- Frequenzgang, Bode-Diagramm
- Minimalphasige und nichtminimalphasige Systeme
- Nyquist-Diagramm, Nyquist-Stabilitätskriterium, Phasenreserve und Amplitudenreserve
- Loop shaping, Lead-Lag-Kompensatoren
- Frequenzgang von PID-Regelkreisen

Totzeitsysteme

- Wurzelortskurve und Frequenzgang von Totzeitsystemen
- Smith-Prädiktor

Digitale Regelung

- Abtastsysteme, Differenzengleichungen
- Tustin-Approximation, digitale PID-Regler

Software-Werkzeuge

- Einführung in Matlab, Simulink, Control Toolbox
- Rechnergestützte Aufgaben zu allen Themen der Vorlesung

Literatur:

- Werner, H., Lecture Notes „Introduction to Control Systems“
 - G.F. Franklin, J.D. Powell and A. Emami-Naeini "Feedback Control of Dynamic Systems", Addison Wesley, Reading, MA, 2009
 - K. Ogata "Modern Control Engineering", Fourth Edition, Prentice Hall, Upper Saddle River, NJ, 2010
 - R.C. Dorf and R.H. Bishop, "Modern Control Systems", Addison Wesley, Reading, MA 2010
-

Lehrveranstaltung: Grundlagen der Regelungstechnik (Gruppenübung)

Dozenten:

Prof. Herbert Werner

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Numerik und Computer Algebra	Vorlesung	2
Numerik und Computer Algebra	Seminar	2
Numerik und Computer Algebra	Gruppenübung	1

Modulverantwortlich:

Prof. Siegfried Rump

Zulassungsvoraussetzung:

keine

Empfohlene Vorkenntnisse:

Grundkenntnisse in numerischer und diskreter Mathematik

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden kennen den Unterschied zwischen Rechengenauigkeit und Ergebnisgenauigkeit. Für diverse, grundlegende Problemstellungen kennen sie approximative sowie exakte Lösungsmöglichkeiten. Sie können zwischen effizient, nicht effizient und prinzipiell unlösbaren Problemen unterscheiden.

Fertigkeiten:

Die Studierenden können komplexe Problemstellungen aus der Mathematik und Informatik analysieren und insbesondere die Empfindlichkeit der Lösung bestimmen. Sie können für diverse Probleme bestmögliche Algorithmen im Hinblick auf die Genauigkeit der Lösung entwerfen.

Personale Kompetenzen:

Sozialkompetenz:

Die Studierenden können in kleinen Gruppen fachspezifische Aufgaben gemeinsam bearbeiten und Ergebnisse in geeigneter Weise präsentieren, zum Beispiel während Kleingruppenübungen.

Selbstständigkeit:

Die Studierenden sind in der Lage, die notwendigen Informationen aus den angegebenen Literaturquellen zu beschaffen und in den Kontext der Vorlesung zu setzen. Sie können ihren Wissensstand mit Hilfe vorlesungsbegleitender Maßnahmen (Quiz-Fragen in den Vorlesungen, klausurnahe Aufgaben) kontinuierlich überprüfen und auf dieser Basis ihre Lernprozesse steuern.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 110, Präsenzstudium: 70

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
 Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
 Informatik-Ingenieurwesen: Vertiefung Ingenieurwesen: Wahlpflicht
 Informatik-Ingenieurwesen myTrack: Vertiefung Ingenieurwesen: Wahlpflicht
 Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Numerik und Computer Algebra (Vorlesung)

Dozenten:

Prof. Siegfried Rump

Sprachen:

DE

Zeitraum:

WS

Inhalt:

- Grundlegende numerische Methoden
- Algorithmen
- Gleitpunktarithmetik IEEE 754
- Arithmetik von Sunaga (Avizienis), Olver, Matula
- Kettenbrüche
- Basic Linear Algebra Subroutines (BLAS)
- Methoden der Computer Algebra
- Turing Maschinen und Berechenbarkeit

Modulhandbuch - Bachelor of Science "Computer Science"

- Churchsche These
- Busy Beaver Funktion
- NP Klassen
- Handlungsreisendenproblem

Literatur:

Higham, N.J.: Accuracy and stability of numerical algorithms, SIAM Publications, Philadelphia, 2nd edition, 2002
Golub, G.H. and Van Loan, Ch.: Matrix Computations, John Hopkins University Press, 3rd edition, 1996
Knuth, D.E.: The Art of Computer Programming: Seminumerical Algorithms, Vol. 2. Addison Wesley, Reading, Massachusetts, 1969

Lehrveranstaltung: Numerik und Computer Algebra (Seminar)

Dozenten:

Prof. Siegfried Rump

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Literatur:

Lehrveranstaltung: Numerik und Computer Algebra (Gruppenübung)

Dozenten:

Prof. Siegfried Rump

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Kombinatorische Strukturen und Algorithmen	Vorlesung	3
Kombinatorische Strukturen und Algorithmen	Gruppenübung	1

Modulverantwortlich:

Prof. Anusch Taraz

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

- Mathematik I + II
- Diskrete Algebraische Strukturen
- Graphentheorie und Optimierung

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- 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.

Fertigkeiten:

- Studierende können Aufgabenstellungen aus der Kombinatorik und Algorithmik mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.
- Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbstständig zu erschließen und können diese verifizieren.
- Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.

Personale Kompetenzen:

Sozialkompetenz:

- 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.

Selbstständigkeit:

- 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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
Technomathematik: Vertiefung Mathematik: Wahlpflicht

Prof. Anusch Taraz

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

- Zählprobleme
- Strukturelle Graphentheorie
- Analyse von Algorithmen
- Extremale Kombinatorik
- Zufällige diskrete Strukturen

Literatur:

- 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: Kombinatorische Strukturen und Algorithmen (Gruppenübung)

Dozenten:

Prof. Anusch Taraz

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Algebraische Methoden in der Regelungstechnik	Vorlesung	2
Algebraische Methoden in der Regelungstechnik	Gruppenübung	2

Modulverantwortlich:

Dr. Prashant Batra

Zulassungsvoraussetzung:

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:

Wissen:

Studierende können

- Input-Output-Systeme polynomial beschreiben,
- Faktorisierungsansätze für Übertragungsfunktionen erklären,
- Stabilisierungsbedingungen für Systeme in coprimer stabiler Faktorisierung benennen.

Fertigkeiten:

Studierende sind in der Lage

- 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:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
Elektrotechnik: Kernqualifikation: Wahlpflicht
Elektrotechnik myTrack: Kernqualifikation: Wahlpflicht
Informatik-Ingenieurwesen: Vertiefung Ingenieurwesen: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Ingenieurwesen: Wahlpflicht
Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Algebraische Methoden in der Regelungstechnik (Vorlesung)

Dozenten:

Dr. Prashant Batra

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

- Algebraische Methoden der Regelungstechnik, polynomialer Ansatz, Faktorisierungsbeschreibung
- Beschreibung 1-dimensionalen Regelsysteme, Synthese von (minimalen) Regelsystemen durch algebraische Interpolationsmethoden,

- Simultane Stabilisierbarkeit
- Parametrisierung sämtlicher stabilisierenden Regler
- Reglerentwurf bei Polvorgabe
- Berücksichtigung von Systemeigenschaften: Störanfälligkeit, Sensitivität.
- Polynomiale Matrizen, Beschreibung durch Links-Faktorisierungen.
- Euklidischer Algorithmus u. Diophantische Gleichungen über Ringen
- Smith-McMillan Normal Form
- Synthese von Mehrgrößensystemen durch polynomiale Methoden

Literatur:

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: Algebraische Methoden in der Regelungstechnik (Gruppenübung)

Dozenten:

Dr. Prashant Batra

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Löser für schwachbesetzte lineare Gleichungssysteme	Vorlesung	2
Löser für schwachbesetzte lineare Gleichungssysteme	Gruppenübung	2

Modulverantwortlich:

Prof. Sabine Le Borne

Zulassungsvoraussetzung:

- Mathematik I + II für Ingenieurstudierende (deutsch oder englisch)

oder

- Analysis & Lineare Algebra I + II für Technomathematiker

Empfohlene Vorkenntnisse:

- Vorlesungsinhalte der Veranstaltungen der Zulassungsvoraussetzungen
- Programmierkenntnisse in C

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Studierende können

- klassische und moderne Iterationsverfahren und deren Zusammenhänge untereinander benennen,
- Konvergenzaussagen zu Iterationsverfahren wiedergeben,
- Aspekte der effizienten Implementierung von Iterationsverfahren erklären.

Fertigkeiten:

Studierende sind in der Lage,

- Iterationsverfahren zu implementieren, anzuwenden und zu vergleichen,
- das Konvergenzverhalten von Iterationsverfahren zu analysieren und gegebenenfalls Konvergenzraten zu berechnen.

Personale Kompetenzen:

Sozialkompetenz:

Studierende können

- in heterogen zusammengesetzten Teams (d.h. aus unterschiedlichen Studiengängen und mit unterschiedlichem Hintergrundwissen) zusammenarbeiten, sich theoretische Grundlagen erklären sowie bei praktischen Implementierungsaspekten der Algorithmen unterstützen.

Selbstständigkeit:

Studierende sind fähig,

- selbst einzuschätzen, ob sie die begleitenden theoretischen und praktischen Übungsaufgaben besser allein oder im Team lösen,
- mit ausreichender Ausdauer komplexe Problemstellungen über längere Zeiträume zu bearbeiten,
- ihren Lernstand konkret zu beurteilen und gegebenenfalls gezielt Fragen zu stellen und Hilfe zu suchen.

Leistungspunkte:

6 LP

Prüfung:

Mündliche Prüfung

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computational Mathematics: Wahlpflicht
Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
Elektrotechnik: Kernqualifikation: Wahlpflicht
Elektrotechnik myTrack: Kernqualifikation: Wahlpflicht
Informatik-Ingenieurwesen: Vertiefung Ingenieurwesen: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Ingenieurwesen: Wahlpflicht
Technomathematik: Vertiefung Mathematik: Wahlpflicht

Lehrveranstaltung: Löser für schwachbesetzte lineare Gleichungssysteme (Vorlesung)

Dozenten:

Prof. Sabine Le Borne

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

1. Schwachbesetzte Matrizen: Anordnungen und Speicherformate, direkte Löser
2. Klassische Iterationsverfahren: Grundbegriffe, Konvergenz
3. Projektionsverfahren
4. Krylovraumverfahren
5. Präkonditionierung (z.B. ILU)
6. Mehrgitterverfahren

Literatur:

1. Y. Saad, Iterative methods for sparse linear systems
-

Lehrveranstaltung: Löser für schwachbesetzte lineare Gleichungssysteme (Gruppenübung)

Dozenten:

Prof. Sabine Le Borne

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Mathematik IV

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Differentialgleichungen 2 (Partielle Differentialgleichungen)	Vorlesung	2
Differentialgleichungen 2 (Partielle Differentialgleichungen)	Gruppenübung	1
Differentialgleichungen 2 (Partielle Differentialgleichungen)	Hörsaalübung	1
Komplexe Funktionen	Vorlesung	2
Komplexe Funktionen	Gruppenübung	1
Komplexe Funktionen	Hörsaalübung	1

Modulverantwortlich:

Prof. Anusch Taraz

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Mathematik I - III

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- Studierende können die grundlegenden Begriffe der Mathematik IV 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.

Fertigkeiten:

- Studierende können Aufgabenstellungen aus der Mathematik IV mit Hilfe der kennengelernten Konzepte modellieren und mit den erlernten Methoden lösen.
- Studierende sind in der Lage, sich weitere logische Zusammenhänge zwischen den kennengelernten Konzepten selbständig zu erschließen und können diese verifizieren.
- Studierende können zu gegebenen Problemstellungen einen geeigneten Lösungsansatz entwickeln, diesen verfolgen und die Ergebnisse kritisch auswerten.

Personale Kompetenzen:

Sozialkompetenz:

- 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.

Selbstständigkeit:

- 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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 68, Präsenzstudium: 112

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Elektrotechnik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Schiffbau: Pflicht
Computer Science: Vertiefung Computational Mathematics: Wahlpflicht

Modulhandbuch - Bachelor of Science "Computer Science"

Computer Science myTrack: Vertiefung Computational Mathematics: Wahlpflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Elektrotechnik: Pflicht
General Engineering Science: Vertiefung Schiffbau: Pflicht
General Engineering Science: Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht
General Engineering Science: Vertiefung Maschinenbau, Schwerpunkt Theoretischer Maschinenbau: Pflicht
Informatik-Ingenieurwesen: Vertiefung Ingenieurwesen: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Ingenieurwesen: Wahlpflicht
Maschinenbau: Vertiefung Theoretischer Maschinenbau: Pflicht
Maschinenbau: Vertiefung Mechatronik: Pflicht
Mechatronik: Kernqualifikation: Pflicht
Schiffbau: Kernqualifikation: Pflicht
Theoretischer Maschinenbau: Technischer Ergänzungskurs Kernfächer: Wahlpflicht

Lehrveranstaltung: Differentialgleichungen 2 (Partielle Differentialgleichungen) (Vorlesung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Grundzüge der Theorie und Numerik partieller Differentialgleichungen

- Beispiele für partielle Differentialgleichungen
- quasilineare Differentialgleichungen erster Ordnung
- Normalformen linearer Differentialgleichungen zweiter Ordnung
- harmonische Funktionen und Maximumprinzip
- Maximumprinzip für die Wärmeleitungsgleichung
- Wellengleichung
- Lösungsformel nach Liouville
- spezielle Funktionen
- Differenzenverfahren
- finite Elemente

Literatur:

- R. Ansorge, H. J. Oberle: Mathematik für Ingenieure, Band 2; Verlag Wiley-VCH, Berlin, Weinheim, New York, 2000
 - P. Henrici, R. Jelsch: Komplexe Analysis für Ingenieure, Birkhäuser Verlag, Basel, 1998
 - A. Tveit, R. Winther: Einführung in partielle Differentialgleichungen, Springer Verlag, Berlin, Heidelberg, New York, 2002
-

Lehrveranstaltung: Differentialgleichungen 2 (Partielle Differentialgleichungen) (Gruppenübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Differentialgleichungen 2 (Partielle Differentialgleichungen) (Hörsaalübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Komplexe Funktionen (Vorlesung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Grundzüge der Funktionentheorie

- Funktionen einer komplexen Variable
- Komplexe Differentiation
- Konforme Abbildungen
- Komplexe Integration
- Cauchyscher Hauptsatz
- Cauchysche Integralformel
- Taylor- und Laurent-Reihenentwicklung
- Singularitäten und Residuen
- Integraltransformationen: Fourier und Laplace-Transformation

Literatur:

- R. Ansorge, H. J. Oberle: Mathematik für Ingenieure, Band 2; Verlag Wiley-VCH, Berlin, Weinheim, New York, 2000
 - P. Henrici, R. Jelsch: Komplexe Analysis für Ingenieure, Birkhäuser Verlag, Basel, 1998
-

Lehrveranstaltung: Komplexe Funktionen (Gruppenübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltung: Komplexe Funktionen (Hörsaalübung)

Dozenten:

Dozenten des Fachbereiches Mathematik der UHH

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Fachmodule der Vertiefung Computer Engineering

Modul: Grundlagen der Informatik

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Grundlagen der Informatik	Vorlesung	2
Grundlagen der Informatik	Gruppenübung	2

Modulverantwortlich:

Prof. Karl-Heinz Zimmermann

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Keine Vorkenntnisse notwendig.

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden kennen

- Stellenwertsysteme,
- Boolesche Funktionen und Schaltnetze,
- Aufbau, Organisation und Arbeitsweise des von-Neumann-Rechners,
- Assembler und Maschinenprogramme,
- Grundzüge der blockstrukturierten Programmierung.

Fertigkeiten:

Die Studierenden können

- Stellenwertsysteme beschreiben und in Stellenwertsystemen rechnen,
- Boolesche Funktionen spezifizieren und analysieren,
- kombinatorische Schaltungen (Rechenwerk) entwerfen,
- den Ablauf von Programmen in einem von-Neumann-Rechner beschreiben,
- in Assembler einfache Beispiele programmieren,
- in einer blockstrukturierten Sprache einfache Beispiele programmieren,
- algorithmisch denken.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht
 Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
 Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
 Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Grundlagen der Informatik (Vorlesung)

Dozenten:

NN

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

Literatur:

Lehrveranstaltung: Grundlagen der Informatik (Gruppenübung)

Dozenten:

NN

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Einführung in die Nachrichtentechnik und ihre stochastischen Methoden	Vorlesung	3
Einführung in die Nachrichtentechnik und ihre stochastischen Methoden	Hörsaalübung	1

Modulverantwortlich:

Prof. Gerhard Bauch

Zulassungsvoraussetzung:

Mathematik 1-3
Signale und Systeme

Empfohlene Vorkenntnisse:

Grundkenntnisse in Wahrscheinlichkeitsrechnung

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden kennen und verstehen die grundlegenden Funktionseinheiten eines Nachrichtenübertragungssystems. Sie können die einzelnen Funktionsblöcke mit Hilfe grundlegender Kenntnisse der Signal- und Systemtheorie sowie der Theorie stochastischer Prozesse beschreiben und analysieren. Sie kennen die entscheidenden Ressourcen und Bewertungskriterien der Nachrichtenübertragung und können ein elementares nachrichtentechnisches System entwerfen und beurteilen.

Fertigkeiten:

Die Studierenden sind in der Lage, ein elementares nachrichtentechnisches System zu entwerfen und zu beurteilen. Insbesondere können Sie den Bedarf an Ressourcen wie Bandbreite und Leistung abschätzen. Sie sind in der Lage, wichtige Beurteilungskriterien wie die Bandbreiteneffizienz oder die Bitfehlerwahrscheinlichkeit elementarer Nachrichtenübertragungssysteme abzuschätzen und darauf basierend ein Übertragungsverfahren auszuwählen.

Personale Kompetenzen:

Sozialkompetenz:

Die Studierenden können in fachspezifische Aufgaben gemeinsam bearbeiten.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Elektrotechnik: Pflicht
Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Elektrotechnik: Pflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Einführung in die Nachrichtentechnik und ihre stochastischen Methoden (Vorlesung)

Dozenten:

Prof. Gerhard Bauch

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

- Grundlagen stochastischer Prozesse
- Einführung in die Nachrichtentechnik
- Quadraturamplitudenmodulation
- Beschreibung hochfrequenter Nachrichtenübertragung im äquivalenten Basisband

Modulhandbuch - Bachelor of Science "Computer Science"

- Übertragungskanäle, Kanalmodelle
- Analog-Digital-Wandlung: Abtastung, Quantisierung, Pulsecodemodulation (PCM)
- Grundlagen der Informationstheorie, Quellencodierung und Kanalcodierung
- Digitale Basisbandübertragung: Pulsformung, Augendiagramm, 1. und 2. Nyquist-Bedingung, Matched-Filter, Detektion, Fehlerwahrscheinlichkeit
- Grundlagen digitaler Modulationsverfahren

Literatur:

K. Kammeyer: Nachrichtenübertragung, Teubner
P.A. Höher: Grundlagen der digitalen Informationsübertragung, Teubner.
M. Bossert: Einführung in die Nachrichtentechnik, Oldenbourg.
J.G. Proakis, M. Salehi: Grundlagen der Kommunikationstechnik. Pearson Studium.
J.G. Proakis, M. Salehi: Digital Communications. McGraw-Hill.
S. Haykin: Communication Systems. Wiley
J.G. Proakis, M. Salehi: Communication Systems Engineering. Prentice-Hall.
J.G. Proakis, M. Salehi, G. Bauch, Contemporary Communication Systems. Cengage Learning.

Lehrveranstaltung: Einführung in die Nachrichtentechnik und ihre stochastischen Methoden (Hörsaalübung)

Dozenten:

Prof. Gerhard Bauch

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Elektrotechnisches Versuchspraktikum	Laborpraktikum	2
Messtechnik und Messdatenverarbeitung	Vorlesung	2
Messtechnik und Messdatenverarbeitung	Gruppenübung	1

Modulverantwortlich:

Prof. Alexander Schlaefer

Zulassungsvoraussetzung:

keine

Empfohlene Vorkenntnisse:

Grundlagen Mathematik

Grundlagen Elektrotechnik

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Die Studierenden können die Aufgaben von Messsystemen sowie das Vorgehen bei der Messdatenerfassungen und -verarbeitungen erklären. Die für die Messtechnik relevanten Aspekte der Wahrscheinlichkeitstheorie und der Messfehlerbehandlung sowie das Vorgehen bei der Messungen stochastischer Signale können wiedergegeben werden. Methoden zur Beschreibungen gemessener Signale und zur Digitalisierungen von Signalen sind den Studierenden bekannt und können erläutert werden.

Fertigkeiten:

Die Studierenden sind in der Lage messtechnische Fragestellungen zu erklären und Methoden zur Beschreibung und Verarbeitung von Messdaten anzuwenden.

Personale Kompetenzen:

Sozialkompetenz:

Die Studierenden lösen Übungsaufgaben in Kleingruppen.

Selbstständigkeit:

Die Studierenden können ihren Wissensstand einschätzen und die von Ihnen erzielten Ergebnisse kritisch bewerten.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 110, Präsenzstudium: 70

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Elektrotechnik: Pflicht
Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Elektrotechnik: Pflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Elektrotechnisches Versuchspraktikum (Laborpraktikum)

Dozenten:

Prof. Alexander Schlaefer, Prof. Christian Schuster, Prof. Günter Ackermann, Prof. Rolf-Rainer Grigat, Prof. Arne Jacob, Prof. Herbert Werner, Dozenten des SD E, Prof. Heiko Falk

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Praktikumsversuche
"Digitale Schaltungen" Prof. Grigat
"Halbleiter-Bauelemente" Prof. Jacob

Modulhandbuch - Bachelor of Science "Computer Science"

"Mikrocontroller" Prof. Mayer-Lindenb.
"Analoge Schaltungen" Prof. Werner
"Leistung im Wechselstromkreis" Prof. Schuster
"Elektrische Maschinen" Prof. Ackermann

Literatur:

Wird in der Lehrveranstaltung festgelegt

Lehrveranstaltung: Messtechnik und Messdatenverarbeitung (Vorlesung)

Dozenten:

Prof. Alexander Schlaefer

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Einführung, Messsysteme und Messfehler, Wahrscheinlichkeitstheorie, Messung stochastischer Signale, Beschreibung gemessener Signale,
Erfassung analoger Signale, Praktische Messdatenerfassung

Literatur:

Puente León, Kiencke: Messtechnik, Springer 2012
Lerch: Elektrische Messtechnik, Springer 2012

Weitere Literatur wird in der Veranstaltung bekanntgegeben.

Lehrveranstaltung: Messtechnik und Messdatenverarbeitung (Gruppenübung)

Dozenten:

Prof. Alexander Schlaefer

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Rechnerarchitektur

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Rechnerarchitektur	Vorlesung	2
Rechnerarchitektur	Gruppenübung	2

Modulverantwortlich:

Prof. Heiko Falk

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Modul "Technische Informatik"

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

In diesem Modul werden fortgeschrittene Konzepte der Rechnerarchitektur vorgestellt. Am Anfang steht ein breiter Überblick über mögliche Programmiermodelle, wie sie für Universalrechner (z.B. MIPS-Architektur) 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. Im letzten Abschnitt werden Rechensysteme mit mehr als einem Prozessor bzw. mehreren Recheneinheiten betrachtet.

Fertigkeiten:

Die Studierenden sind in der Lage, den Aufbau eines Prozessors zu erklären. Sie kennen die verschiedenen Architekturprinzipien und Programmiermodelle. Die Studierenden untersuchen verschiedene Strukturen der MIPS-Architektur und sind in der Lage, Pipeline-Konflikte zu erklären und zu analysieren. Sie bewerten unterschiedliche Speicherarchitekturen, kennen parallele Rechnerarchitekturen und können zwischen Befehls- und Datenparallelität unterscheiden.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

Die Studierenden sind nach Abschluss des Moduls in der Lage, sich Teilbereiche des Fachgebietes anhand von Fachliteratur selbständig zu erarbeiten, das erworbene Wissen zusammenzufassen, zu präsentieren und es mit den Inhalten anderer Lehrveranstaltungen zu verknüpfen.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Rechnerarchitektur (Vorlesung)

Dozenten:

Prof. Heiko Falk

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

In dieser Vorlesung werden fortgeschrittene Konzepte der Rechnerarchitektur vorgestellt. Am Anfang steht ein breiter Überblick über mögliche Programmiermodelle, wie sie für Universalrechner (z.B. MIPS-Architektur) aber auch 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 angewendeten Methoden zur Beschleunigung der Befehlsausführung.

Gliederung der Vorlesung:

- Einführung
- Grundlagen von VHDL
- Programmiermodelle
- Der CO₂-Fußabdruck von PCs
- Realisierung elementarer Datentypen
- Dynamisches Scheduling
- Sprungvorhersage
- Superskalare Maschinen
- Speicher-Hierarchien

Literatur:

- D. Patterson, J. Hennessy. Rechnerorganisation und -entwurf. Elsevier, 2005.
 - A. Tanenbaum, J. Goodman. Computerarchitektur. Pearson, 2001.
-

Lehrveranstaltung: Rechnerarchitektur (Gruppenübung)

Dozenten:

Prof. Heiko Falk

Sprachen:

DE/EN

Zeitraum:

WS

Inhalt:

Literatur:

Modul: Verteilte Systeme

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Verteilte Systeme	Vorlesung	2
Verteilte Systeme	Gruppenübung	2

Modulverantwortlich:

Prof. Volker Turau

Zulassungsvoraussetzung:

- Prozedurales Programmieren
- Objektorientiertes Programmieren mit Java
- Rechnernetze

Empfohlene Vorkenntnisse:

- Socket Programmierung

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Studierende können die wichtigsten Abstraktion von Verteilten Systemen erklären (Marshalling, Proxy, Dienst, Adresse, Entfernter Aufruf, synchrones/asynchrones System). Sie sind in der Lage, die Vor- und Nachteile verschiedener Arten von Interprozesskommunikation zu beschreiben. Sie kennen die wichtigsten Architekturvarianten von Verteilten Systemen einschließlich ihrer Vor- und Nachteile. Die Teilnehmer sind in der Lage, mindestens drei Synchronisationsverfahren zu beschreiben.

Fertigkeiten:

Studierende können auf unterschiedliche Arten verteilte Systeme realisieren. Dabei können sie folgende Methoden verwenden:

- Eigenes Protokoll entwerfen und mittels TCP umsetzen
- HTTP als entfernten Aufruf nutzen
- RMI als Middleware nutzen

Personale Kompetenzen:

Sozialkompetenz:

Selbstständigkeit:

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht
 Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
 Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
 Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
 Technomathematik: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Verteilte Systeme (Vorlesung)

Dozenten:

Prof. Volker Turau

Sprachen:

DE

Zeitraum:

WS

Inhalt:

- Architekturen für verteilte Systeme
- HTTP: Einfacher entfernter Aufruf
- Client-Server Architekturen
- Entfernter Aufruf
- Remote Method Invocation (RMI)

- Synchronisierung
- Verteiltes Caching
- Namensdienste
- Verteilte Dateisysteme

Literatur:

- Verteilte Systeme – Prinzipien und Paradigmen, Andrew S. Tanenbaum, Maarten van Steen, Pearson Studium
 - Verteilte Systeme, G. Coulouris, J. Dollimore, T. Kindberg, 2005, Pearson Studium
-

Lehrveranstaltung: Verteilte Systeme (Gruppenübung)

Dozenten:

Prof. Volker Turau

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Quantenmechanik für Studierende der Ingenieurwissenschaften	Vorlesung	2
Quantenmechanik für Studierende der Ingenieurwissenschaften	Gruppenübung	2

Modulverantwortlich:

Prof. Wolfgang Hansen

Zulassungsvoraussetzung:

Physik- und Mathematikmodul des Bachelorstudiengangs

Empfohlene Vorkenntnisse:

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:

Wissen:

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.

Fertigkeiten:

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:

Sozialkompetenz:

Die Studierenden diskutieren den Vorlesungsstoff und präsentieren Lösungen einfacher quantenmechanischer Probleme in Kleingruppen während der Übungen.

Selbstständigkeit:

Die Studierenden sind in der Lage selbstständig einfache Lösungswege zu quantenmechanischen 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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Elektrotechnik: Kernqualifikation: Wahlpflicht
Informatik-Ingenieurwesen: Vertiefung Ingenieurwesen: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Ingenieurwesen: Wahlpflicht

Lehrveranstaltung: Quantenmechanik für Studierende der Ingenieurwissenschaften (Vorlesung)

Dozenten:

Prof. Wolfgang Hansen

Sprachen:

DE

Zeitraum:

WS

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 Tunneliode, Bandstruktur, Zustandsdichte, Besetzungsverteilung, Zener-Diode, Fermis Goldene Regel und Übergangsmatrixelemente, Heterostrukturlaser, Quantenkaskadenlaser

Literatur:

Autor	Titel	Verlag	ISBN-Nr.	Jahr
-------	-------	--------	----------	------

Modulhandbuch - Bachelor of Science "Computer Science"

David K. Ferry	Quantum Mechanics	IOP Publishing Ltd	0-7503-0327-1 (hbk) 1995 0-7503-0328-X (pbk)
M. Jaros	Physics and Applications of Semiconductor Microstructures	Clarendon Press	0-19-851994-X 1989 0-19-853927-4 (Pbk)
Randy Harris	Moderne Physik Lehr- und Übungsbuch 2., aktualisierte Auflage Kapitel 3-10	Pearson Deutschland GmbH	978-3-86894-115-9 2013

Lehrveranstaltung: Quantenmechanik für Studierende der Ingenieurwissenschaften (Gruppenübung)

Dozenten:

Prof. Wolfgang Hansen

Sprachen:

DE

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Databases

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Datenbanken	Vorlesung	4
Datenbanken	Problemorientierte Lehrveranstaltung	1

Modulverantwortlich:

NN

Zulassungsvoraussetzung:

Empfohlene Vorkenntnisse:

Students should have basic knowledge in the following areas:

- Discrete Algebraic Structures
- Procedural Programming
- Logic, Automata, and Formal Languages
- Object-Oriented Programming, Algorithms and Data Structures

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Students can explain the general architecture of an application system that is based on a database. They describe the syntax and semantics of the Entity Relationship conceptual modeling languages, and they can enumerate basic decision problems and know which features of a domain model can be captured with ER and which features cannot be represented. Furthermore, students can summarize the features of the relational data model, and can describe how ER models can be systematically transformed into the relational data model. Student are able to discuss dependency theory using the operators of relational algebra, and they know how to use relational algebra as a query language. In addition, they can sketch the main modules of the architecture of a database system from an implementation point of view. Storage and index structures as well as query answering and optimization techniques can be explained. The role of transactions can be described in terms of ACID conditions and common recovery mechanisms can be characterized. The students can recall why recursion is important for query languages and describe how Datalog can be used and implemented. They demonstrate how Datalog can be used for information integration. For solving ER decision problems the students can explain description logics with their syntax and semantics, they describe description logic decision problems and explain how these problems can be mapped onto each other. They can sketch the idea of ontology-based data access and can name the main complexity measure in database theory. Last but not least, the students can describe the main features of XML and can explain XPath and XQuery as query languages.

Fertigkeiten:

Students can apply ER for describing domains for which they receive a textual description, and students can transform relational schemata with a given set of functional dependencies into third normal form or even Boyce-Codd normal form. They can also apply relational algebra, SQL, or Datalog to specify queries. Using specific datasets, they can explain how index structures work (e.g., B-trees) and how index structures change while data is added or deleted. They can rewrite queries for better performance of query evaluation. Students can analyse which query language expressivity is required for which application problem. Description logics can be applied for domain modeling, and students can transform ER diagrams into description logics in order to check for consistency and implicit subsumption relations. They solve data integration problems using Datalog and LAV or GAV rules. Students can apply XPath and Xquery to retrieve certain patterns in XML data.

Personale Kompetenzen:

Sozialkompetenz:

Students develop an understanding of social structures in a company used for developing real-world products. They know the responsibilities of data analysts, programmers, and managers in the overall production process.

Selbstständigkeit:

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 110, Präsenzstudium: 70

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Computer Science: Vertiefung Computer Engineering: Wahlpflicht
 Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
 General Engineering Science: Vertiefung Informatik-Ingenieurwesen: Pflicht
 Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
 Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
 Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Databases (Vorlesung)

Dozenten:

NN

Sprachen:

EN

Zeitraum:

WS

Inhalt:

- 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:

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: Databases (Problemorientierte Lehrveranstaltung)

Dozenten:

NN

Sprachen:

EN

Zeitraum:

WS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Einführung in Medizintechnische Systeme	Vorlesung	2
Einführung in Medizintechnische Systeme	Problemorientierte Lehrveranstaltung	4

Modulverantwortlich:

Prof. Alexander Schlaefer

Zulassungsvoraussetzung:

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:

Wissen:

Die Studierenden können die Funktionen von medizintechnischen Systemen wie beispielsweise bildgebenden Systemen, Assistenzsystemen im OP, medizintechnischen Sensorsystemen und medizintechnischen Informationssystemen erklären. Sie können einen Überblick über Regulatorische Rahmenbedingungen und Standards in der Medizintechnik geben.

Fertigkeiten:

Die Studierenden sind in der Lage ihr grundlegendes Verständnis von medizintechnischen Systemen auf praxisrelevante Problemstellungen anzuwenden.

Personale Kompetenzen:

Sozialkompetenz:

Die Studierenden können in Gruppen ein medizintechnisches Thema als Projekt beschreiben, in Teilaufgaben untergliedern und gemeinsam bearbeiten.

Selbstständigkeit:

Die Studierenden können ihren Wissensstand einschätzen und ihre Arbeitsergebnisse dokumentieren. Sie können die erzielten Ergebnisse kritisch bewerten und in geeigneter Weise präsentieren.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 96, Präsenzstudium: 84

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Mediziningenieurwesen: Pflicht
Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Elektrotechnik: Kernqualifikation: Wahlpflicht
Elektrotechnik myTrack: Kernqualifikation: Wahlpflicht
General Engineering Science: Vertiefung Mediziningenieurwesen: Pflicht
Mediziningenieurwesen: Vertiefung Künstliche Organe und Regenerative Medizin: Wahlpflicht
Mediziningenieurwesen: Vertiefung Implantate und Endoprothesen: Wahlpflicht
Mediziningenieurwesen: Vertiefung Medizin- und Regelungstechnik: Wahlpflicht
Mediziningenieurwesen: Vertiefung Management und Administration: Wahlpflicht
Technomathematik: Vertiefung Ingenieurwissenschaften: Wahlpflicht

Lehrveranstaltung: Einführung in Medizintechnische Systeme (Vorlesung)

Dozenten:

Prof. Alexander Schlaefer

Sprachen:

DE

Zeitraum:

SS

Inhalt:

- Bildgebende Systeme
- Assistenzsysteme im OP

Modulhandbuch - Bachelor of Science "Computer Science"

- 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: Einführung in Medizintechnische Systeme (Problemorientierte Lehrveranstaltung)

Dozenten:

Prof. Alexander Schlaefer

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Compiler Construction

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Compilerbau	Vorlesung	2
Compilerbau	Gruppenübung	2

Modulverantwortlich:

Prof. Sibylle Schupp

Zulassungsvoraussetzung:

- Automata theory and formal languages
- Functional programming or procedural programming
- Object-oriented programming, algorithms, and data structures

Empfohlene Vorkenntnisse:

- Practical programming experience
- Basic knowledge of software engineering

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

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.

Fertigkeiten:

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:

Sozialkompetenz:

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.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Projektarbeit

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht

Lehrveranstaltung: Compiler Construction (Vorlesung)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

SS

Inhalt:

- Lexical and syntactic analysis
- Semantic analysis
- Code generation
- Optimization
- Run-time support

Literatur:

Alfred Aho, Jeffrey Ullman, Ravi Sethi, and Monica S. Lam, Compilers: Principles, Techniques, and Tools, 2nd edition

Lehrveranstaltung: Compiler Construction (Gruppenübung)

Dozenten:

Prof. Sibylle Schupp

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Application Security

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Anwendungssicherheit	Vorlesung	3
Anwendungssicherheit	Gruppenübung	2

Modulverantwortlich:

Prof. Dieter Gollmann

Zulassungsvoraussetzung:

None

Empfohlene Vorkenntnisse:

Familiarity with Information security, fundamentals of cryptography, Web protocols and the architecture of the Web

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Students can name current approaches for securing selected applications, in particular of web applications

Fertigkeiten:

Students are capable of

- performing a security analysis
- developing security solutions for distributed applications
- recognizing the limitations of existing standard solutions

Personale Kompetenzen:

Sozialkompetenz:

Students are capable of appreciating the impact of security problems on those affected and of the potential responsibilities for their resolution.

Selbstständigkeit:

Students are capable of acquiring knowledge independently from professional publications, technical standards, and other sources, and are capable of applying newly acquired knowledge to new problems.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 110, Präsenzstudium: 70

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht

Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht

Informatik-Ingenieurwesen: Vertiefung Informations- und Kommunikationstechnik: Wahlpflicht

Information and Communication Systems: Vertiefung Kommunikationssysteme, Schwerpunkt Software: Wahlpflicht

Information and Communication Systems: Vertiefung Sichere und zuverlässige IT-Systeme: Wahlpflicht

Internationales Wirtschaftsingenieurwesen: Vertiefung II. Informationstechnologie: Wahlpflicht

Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Application Security (Vorlesung)

Dozenten:

Prof. Dieter Gollmann

Sprachen:

EN

Zeitraum:

SS

Inhalt:

- Email security
- Web Services security
- Security in Web applications
- Access control
- Trust Management

- Trusted Computing
- Digital Rights Management
- Security Solutions for selected applications

Literatur:

Webseiten der OMG, W3C, OASIS, WS-Security, OECD, TCG
D. Gollmann: Computer Security, 3rd edition, Wiley (2011)
R. Anderson: Security Engineering, 2nd edition, Wiley (2008)
U. Lang: CORBA Security, Artech House, 2002

Lehrveranstaltung: Application Security (Gruppenübung)

Dozenten:

Prof. Dieter Gollmann

Sprachen:

EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Halbleiterschaltungstechnik	Vorlesung	3
Halbleiterschaltungstechnik	Gruppenübung	1

Modulverantwortlich:

Prof. Wolfgang Krautschneider

Zulassungsvoraussetzung:

keine

Empfohlene Vorkenntnisse:

Grundlagen der Elektrotechnik
Elementare Grundlagen der Physik

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- Studierende können die Funktionsweisen von verschiedenen MOS-Bauelementen in unterschiedlichen Schaltungen erklären.
- Studierende sind in der Lage, grundlegende digitale Logik-Schaltungen zu benennen und ihre Vor- und Nachteile zu diskutieren.
- Studierende können aktuelle Speichertypen benennen, deren Funktionsweise erklären und Kenngrößen angeben.
- Studierende können die Funktionsweise von Analogschaltungen und deren Anwendungen erklären.
- Studierende können geeignete Anwendungsbereiche von Bipolartransistoren benennen.

Fertigkeiten:

- Studierende können Kenngrößen von verschiedenen MOS-Bauelementen berechnen und Schaltungen dimensionieren.
- Studierende können logische Schaltungen mit unterschiedlichen Schaltungstypen entwerfen und dimensionieren.
- Studierende können MOS-Bauelemente und Operationsverstärker sowie bipolare Transistoren in speziellen Anwendungsbereichen einsetzen.

Personale Kompetenzen:

Sozialkompetenz:

- Studierende sind in der Lage, in heterogen (aus unterschiedlichen Studiengängen) zusammengestellten Teams zusammenzuarbeiten.
- Studierende können in kleinen Gruppen Rechenaufgaben lösen und Fachfragen beantworten.

Selbstständigkeit:

- Studierende sind in der Lage, ihren eigenen Lernstand einzuschätzen.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Vertiefung Elektrotechnik: Pflicht
Allgemeine Ingenieurwissenschaften: Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht
Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Elektrotechnik: Kernqualifikation: Pflicht
Elektrotechnik myTrack: Kernqualifikation: Pflicht
General Engineering Science: Vertiefung Elektrotechnik: Pflicht
General Engineering Science: Vertiefung Maschinenbau, Schwerpunkt Mechatronik: Pflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
Maschinenbau: Vertiefung Mechatronik: Pflicht
Mechatronik: Kernqualifikation: Pflicht
Technomathematik: Kernqualifikation: Wahlpflicht

Lehrveranstaltung: Halbleiterschaltungstechnik (Vorlesung)

Dozenten:

Prof. Wolfgang Krautschneider

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Inhalt:

- Grundsaltungen mit MOS-Transistoren für Logikgatter und Verstärker
- Typische Anwendungsfälle in der digitalen und analogen Schaltungstechnik
- Realisierung logischer Funktionen
- Schaltungen für die Speicherung von binären Daten
- Strukturverkleinerung von CMOS-Schaltkreisen und weitere Leistungssteigerung
- Operationsverstärker und ihre Anwendungen
- Grundsaltungen mit bipolaren Transistoren
- Dimensionierung beispielhafter Schaltungen
- Elektrisches Verhalten von BICMOS-Schaltungen

Literatur:

R. J. Baker, CMOS - Circuit Design, Layout and Simulation, J. Wiley & Sons Inc., 3. Auflage, 2011, ISBN: 047170055S
H.-G. Wagemann und T. Schönauer, Silizium-Planartechnologie, Grundprozesse, Physik und Bauelemente, Teubner-Verlag, 2003, ISBN 3519004674
K. Hoffmann, Systemintegration, Oldenbourg-Verlag, 2. Aufl. 2006, ISBN: 3486578944
U. Tietze und Ch. Schenk, E. Gamm, Halbleiterschaltungstechnik, Springer Verlag, 14. Auflage, 2012, ISBN 3540428496
H. Göbel, Einführung in die Halbleiter-Schaltungstechnik, Berlin, Heidelberg Springer-Verlag Berlin Heidelberg, 2011, ISBN: 9783642208874 ISBN: 9783642208867
URL: <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10499499>
URL: <http://dx.doi.org/10.1007/978-3-642-20887-4>
URL: http://ebooks.ciando.com/book/index.cfm/bok_id/319955
URL: <http://www.ciando.com/img/bo>

Lehrveranstaltung: Halbleiterschaltungstechnik (Gruppenübung)

Dozenten:

Prof. Wolfgang Krautschneider

Sprachen:

DE

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Modul: Eingebettete Systeme

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Eingebettete Systeme	Vorlesung	3
Eingebettete Systeme	Gruppenübung	1

Modulverantwortlich:

Prof. Heiko Falk

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Technische Informatik

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

Eingebettete Systeme sind Systeme, bei denen eine Informationsverarbeitung in eine Umgebung eingebettet ist. In der Vorlesung werden die Grundzüge solcher Systeme vermittelt. Die Vorlesung behandelt insbesondere eine Einführung in diese Systeme (Begriffsbildung, charakteristische Eigenschaften) und deren Spezifikationssprachen (models of computation, hierarchische Zustandsautomaten, Spezifikation von verteilten Systemen, Task-Graphen, Spezifikation von Realzeit-Anwendungen, Übersetzung zwischen Modellen). Ein weiterer Abschnitt behandelt Hardware eingebetteter Systeme: Sensoren, A/D- und D/A-Wandler, realzeitfähige Kommunikationshardware, eingebettete Prozessoren, Speicher, Energiebedarf, rekonfigurierbare Logik und Aktuatoren. Zum Modul gehört auch eine Einführung in Realzeit-Betriebssysteme, Middleware und Realzeit-Scheduling. Schließlich wird auf die Implementierung eingebetteter Systeme mittels Hardware/Software Co-Design (Hardware/Software-Partitionierung, high-level Transformationen der Spezifikation, energieeffiziente Realisierungen, Compiler für Eingebettete Prozessoren) eingegangen.

Fertigkeiten:

Nach dem Besuch der Veranstaltung sollen die Studierenden in der Lage sein, einfache Eingebettete Systeme zu entwickeln. Dabei sollen die Studierenden erkennen können, welche relevanten Bereiche technologischer Kompetenzen eingesetzt werden müssen, um ein funktionierendes Eingebettetes System zu erhalten. Insbesondere sollen sie Modellierungstechniken miteinander vergleichen und geeignete Techniken zur Systementwicklung einsetzen können. Sie sollen beurteilen können, in welchen Bereichen besondere Risiken bestehen.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Klausur

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht
 Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
 Elektrotechnik: Kernqualifikation: Wahlpflicht
 Informatik-Ingenieurwesen: Kernqualifikation: Pflicht
 Informatik-Ingenieurwesen myTrack: Kernqualifikation: Pflicht
 Mechatronics: Vertiefung Systementwurf: Wahlpflicht
 Mechatronics: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht

Lehrveranstaltung: Eingebettete Systeme (Vorlesung)

Dozenten:

Prof. Heiko Falk

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Modulhandbuch - Bachelor of Science "Computer Science"

- Einleitung
- Spezifikation und Modellierung
- Hardware Eingebetteter Systeme
- System-Software
- Evaluation und Validierung
- Abbildung von Anwendungen auf Ausführungsplattformen
- Optimierung

Literatur:

- Peter Marwedel. Embedded System Design - Embedded Systems Foundations of Cyber-Physical Systems. 2nd Edition, Springer, 2012., Springer, 2012.
-

Lehrveranstaltung: Eingebettete Systeme (Gruppenübung)

Dozenten:

Prof. Heiko Falk

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Siehe korrespondierende Vorlesung

Literatur:

Siehe korrespondierende Vorlesung

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
Labor Cyber-Physical Systems	Problemorientierte Lehrveranstaltung	4

Modulverantwortlich:

Prof. Heiko Falk

Zulassungsvoraussetzung:

Keine

Empfohlene Vorkenntnisse:

Modul "Eingebettete Systeme"

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

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.

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 Spezifikationssprachen (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.

Fertigkeiten:

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.

Personale Kompetenzen:

Sozialkompetenz:

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.

Selbstständigkeit:

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.

Leistungspunkte:

6 LP

Prüfung:

Projektarbeit

Arbeitsaufwand in Stunden:

Eigenstudium: 124, Präsenzstudium: 56

Zuordnung zu folgenden Curricula:

Computer Science: Vertiefung Computer Engineering: Wahlpflicht
Computer Science myTrack: Vertiefung Computer Engineering: Wahlpflicht
Informatik-Ingenieurwesen: Vertiefung Informatik: Wahlpflicht
Informatik-Ingenieurwesen myTrack: Vertiefung Informatik: Wahlpflicht
Mechatronik: Vertiefung Intelligente Systeme und Robotik: Wahlpflicht
Mechatronik: Vertiefung Systementwurf: Wahlpflicht

Lehrveranstaltung: Labor Cyber-Physical Systems (Problemorientierte Lehrveranstaltung)

Dozenten:

Prof. Heiko Falk

Sprachen:

DE/EN

Zeitraum:

SS

Inhalt:

Modulhandbuch - Bachelor of Science "Computer Science"

- Versuch 1: Programmieren in NXC
- Versuch 2: Programmierung des Roboters mit Matlab/Simulink
- Programmierung des Roboters in LabVIEW

Literatur:

- Peter Marwedel. Embedded System Design - Embedded System Foundations of Cyber-Physical Systems. 2nd Edition, Springer, 2012.
- Begleitende Foliensätze

Thesis

Modul: Bachelorarbeit

Lehrveranstaltungen:

<u>Titel</u>	<u>Typ</u>	<u>SWS</u>
--------------	------------	------------

Modulverantwortlich:

Professoren der TUHH

Zulassungsvoraussetzung:

Empfohlene Vorkenntnisse:

Modulziele/ angestrebte Lernergebnisse:

Nach erfolgreicher Teilnahme haben die Studierenden die folgenden Lernergebnisse erreicht:

Fachkompetenz:

Wissen:

- 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.

Fertigkeiten:

- 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.

Personale Kompetenzen:

Sozialkompetenz:

- 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.

Selbstständigkeit:

- 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.

Leistungspunkte:

12 LP

Prüfung:

laut FSPO

Arbeitsaufwand in Stunden:

Eigenstudium: 360, Präsenzstudium: 0

Zuordnung zu folgenden Curricula:

Allgemeine Ingenieurwissenschaften: Abschlussarbeit: Pflicht
Bau- und Umweltingenieurwesen: Abschlussarbeit: Pflicht
Bioverfahrenstechnik: Abschlussarbeit: Pflicht
Computer Science: Abschlussarbeit: Pflicht
Computer Science myTrack: Abschlussarbeit: Pflicht
Elektrotechnik: Abschlussarbeit: Pflicht
Elektrotechnik myTrack: Abschlussarbeit: Pflicht
Energie- und Umwelttechnik: Abschlussarbeit: Pflicht
General Engineering Science: Abschlussarbeit: Pflicht
Informatik-Ingenieurwesen: Abschlussarbeit: Pflicht
Informatik-Ingenieurwesen myTrack: Abschlussarbeit: Pflicht
Logistik und Mobilität: Abschlussarbeit: Pflicht
Maschinenbau: Abschlussarbeit: Pflicht
Mechatronik: Abschlussarbeit: Pflicht

Modulhandbuch - Bachelor of Science "Computer Science"

Schiffbau: Abschlussarbeit: Pflicht

Technomathematik: Abschlussarbeit: Pflicht

Verfahrenstechnik: Abschlussarbeit: Pflicht