

**Modulhandbuch  
M.Sc. Informatik  
WS2016/2017-SS2017**

## **Ziele**

Der Master-Studiengang richtet sich an Bachelorabsolventen und setzt somit voraus, dass bereits zu Studienbeginn umfassende Kompetenzen in der Informatik vorhanden sind. Dies schließt breite Kenntnisse und Fertigkeiten im gesamten Gebiet, Erfahrung im Einsatz von Informatikmethoden in Anwendungen und der betrieblichen Praxis, Problemlösungskompetenz, Programmiererfahrung, Erfahrung im Umgang mit abstrakten Modellen sowie Schlüsselqualifikationen ein.

Darauf aufbauend sollen im Masterstudium einerseits die Kenntnisse und Fertigkeiten in der Breite vertieft werden, so dass die Absolventen eine größere Sicherheit beim Einsatz gewinnen und sich im gesamten Bereich der Informatik schnell in neue Techniken einarbeiten können. Andererseits sollen die Absolventen in einem selbst zu wählenden Schwerpunktgebiet an den Stand der Forschung herangeführt werden und tiefgehende wissenschaftlich-methodische Kompetenzen erlangen, die sie befähigen, wissenschaftliche Sachverhalte selbstständig aufzuarbeiten, an der wissenschaftlichen Weiterentwicklung ihres Faches mitzuwirken und ein Promotionsstudium aufzunehmen.

Dabei stellt die Tätigkeit in der Forschung nur eine Option innerhalb eines breiten Tätigkeitsspektrums dar, welches auch die Übernahme leitender Funktionen nach einer Einarbeitungszeit umfasst.

## **Angestrebte Lernergebnisse**

Auf der fachlichen Seite sollen die Absolventen ihre Informatik-Kenntnisse und -Fertigkeiten in der Breite sowie in einem Spezialgebiet vertiefen. Darüber hinaus sollen ihre methodisch-analytischen Kompetenzen, insbesondere die Fähigkeit zur Bearbeitung komplexer Problemstellungen und Auswahl geeigneter Informatikmethoden vertieft werden. Die Studierenden sollen zur wissenschaftlichen Arbeit und leistungsstarke Studierende zur Aufnahme eines Promotionsstudiums befähigt werden. Das Masterstudium bereitet auf die Übernahme leitender Tätigkeiten und anspruchsvoller (Entwicklungs-)Aufgaben vor. Es führt die Studierenden an interdisziplinäre Sicht- und Arbeitsweisen heran und bereitet sie auf Einsatzgebiete im gesamten Spektrum der Informatik und ihrer Anwendungen vor. Dazu gehören insbesondere die Fähigkeit zur sachgerechten und verantwortungsbewussten Anwendung von Informatikmethoden sowie die Fähigkeit zur raschen Einarbeitung in neue Informatik- und Anwendungsgebiete. Masterstudierende erlangen eine Gesamtsicht auf das Fach, die ihnen das Erkennen fachgebietsübergreifender Zusammenhänge ermöglicht.

Neben den fachlichen Qualifikationen vertieft der Masterstudiengang Schlüsselqualifikationen. Großer Wert wird auf die Fähigkeit zur schriftlichen und mündlichen Präsentation, auf Selbstständigkeit, Teamfähigkeit, Kenntnisse und Erfahrungen im Projektmanagement sowie im Rahmen der Möglichkeiten auf persönliche Führungsqualifikationen gelegt. Das Masterstudium fördert die Fähigkeit zur kritischen Reflexion der eigenen Tätigkeit und die Bereitschaft zur Übernahme von Verantwortung für das Ergebnis der eigenen Arbeit. Die Studierenden werden dafür sensibilisiert, bei der Anwendung von Informatiksystemen technische, ökonomische und soziale Randbedingungen zu berücksichtigen.

<b>1. MODULE IM BEREICH VERTIEFUNG IN MATHEMATIK/ELEKTROTECHNIK.....</b>	<b>5</b>
ANALOGUE UND DIGITALE MESSTECHNIK .....	5
COMPUTERALGEBRA 1 .....	7
DIGITAL COMMUNICATION THROUGH BAND-LIMITED CHANNELS .....	8
GRUNDLAGEN DER ALGEBRA UND COMPUTERALGEBRA .....	9
GRUNDLAGEN DER REGELUNGSTECHNIK .....	10
INTRODUCTION TO SIGNAL DETECTION AND ESTIMATION .....	12
INTRODUCTION TO INFORMATION THEORY AND CODING .....	13
KRYPTOGRAPHIE.....	15
MEDIUM ACCESS CONTROL (MAC) PROTOCOLS IN WIRELESS COMMUNICATIONS .....	16
OPTIMIERUNGSVERFAHREN .....	17
PHOTONISCHE KOMPONENTEN UND SYSTEME .....	19
SIMULATION OF DIGITAL COMMUNICATION SYSTEMS USING MATLAB .....	21
STOCHASTIK FÜR INGENIEURE .....	22
<b>2. MODULE IM BEREICH THEORETISCHE INFORMATIK .....</b>	<b>24</b>
AUTOMATEN, SPIELE, LOGIKEN .....	24
DATENBANKTHEORIE.....	26
FORMALE SPRACHEN UND AUTOMATEN I.....	27
KOMPLEXITÄTSTHEORIE .....	28
REDUKTIONSSYSTEME I .....	30
SEMINAR IM BEREICH THEORETISCHE INFORMATIK.....	32
VERIFIKATION EINGEBETTETER SYSTEME .....	33
<b>3. MODULE IM BEREICH TECHNISCHE INFORMATIK.....</b>	<b>34</b>
AUSGEWÄHLTE KAPITEL DER AUTOMATISIERUNG .....	34
AUSGEWÄHLTE KAPITEL DER KOMMUNIKATIONSTECHNIK 2.....	36
AUSGEWÄHLTE KAPITEL DER MIKROPROZESSORTECHNIK .....	37
AUSGEWÄHLTE KAPITEL FÜR PROGRAMMIERSPRACHEN UND TECHNIKEN NACH IEC 61131-3 .....	38
COMMUNICATION TECHNOLOGIES I.....	39
COMMUNICATION TECHNOLOGIES II.....	40
COMPUTATIONAL INTELLIGENCE IN DER AUTOMATISIERUNG .....	41
COMPUTERARITHMETIK/COMPUTER ARITHMETIC .....	43
EINFÜHRUNG IN DIE VIRTUELLE REALITÄT.....	45
JAVA CODE-CAMP CONTEXT AWARENESS 2.....	46
PROZESSRECHNER .....	47
RECHNERGESTÜTZTER ENTWURF MIKROELEKTRONISCHER SCHALTUNGEN.....	48
REKONFIGURIERBARE STRUKTUREN .....	49
RISIKOBEWERTUNG VON RECHNERARCHITEKTUREN 2.....	51
SCHALTUNGSENTWURF MIT HDLS.....	52
SEMINAR IM BEREICH TECHNISCHE INFORMATIK .....	53
SIGNAL- UND BILDVERARBEITUNG .....	54
SYNTHESE UND OPTIMIERUNG MIKROELEKTRONISCHER SYSTEME .....	56
TECHNIK IM BEREICH NEUER MEDIEN .....	57
THEORIE SICHERHEITSGERICHTETER RECHNERSYSTEME .....	58
ZUVERLÄSSIGKEITSTHEORIE FÜR RECHNERSYSTEME .....	59

<b>4. MODULE IM BEREICH PRAKTISCHE INFORMATIK.....</b>	<b>60</b>
GRAPH & MODEL DRIVEN ENGINEERING .....	60
HOME AUTOMATION .....	61
LERNEN IN KOLLABORATIVEN MULTI-AGENTEN SYSTEMEN / LEARNING IN COLLABORATIVE MULTI-AGENT SYSTEMS.....	62
ORGANIC COMPUTING .....	64
PATTERN RECOGNITION .....	66
SEMINAR IM BEREICH PRAKTISCHE INFORMATIK.....	67
SOFTWARE ENGINEERING II.....	68
TEMPORAL AND SPATIAL DATA MINING.....	69
VERTEILTE SYSTEME – BASISALGORITHMEN / DISTRIBUTED COMPUTING ALGORITHMS.....	71
WEB ENGINEERING .....	73
WEB SCIENCE .....	74
<b>5. MODULE IM BEREICH VERTIEFUNG IN THEORETISCHER, TECHNISCHER ODER PRAKTISCHER INFORMATIK .....</b>	<b>75</b>
MODELLGESTÜTZTE FABRIKPLANUNG.....	75
<b>6. SPEZIELLE MODULE.....</b>	<b>77</b>
PROJEKT IN THEORETISCHER, TECHNISCHER ODER PRAKTISCHER INFORMATIK .....	77
SEMINAR IN THEORETISCHER, TECHNISCHER ODER PRAKTISCHER INFORMATIK .....	78
SCHLÜSSELKOMPETENZEN .....	79

# 1. Module im Bereich Vertiefung in Mathematik/Elektrotechnik

Modulbezeichnung:	<i>Analoge und digitale Messtechnik</i>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Lehmann und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundlagen der Elektrotechnik, Analysis, Vorteilhaft: Sensorik- und Matlab-Grundkenntnisse
Angestrebte Lernergebnisse:	Der/die Lernende kann: - sich ein fundiertes Verständnis zeitgemäßer Verfahren der analogen und digitalen Analyse und Verarbeitung von Messsignalen erschließen, - theoretische Kenntnisse durch eigene Programmierübungen überprüfen, - elementare Signal- und Bildverarbeitungsaufgaben analysieren und lösen, - Begriffe und Aufgabenstellungen der Signalverarbeitung in der Messtechnik sicher zuordnen, - Abstraktionsvermögen im Sinne einer systemtheoretischen Denkweise vorweisen, - erworbene Kenntnisse in der Praxis nutzen.
Inhalt:	<u>Teil 1: Analoge Messtechnik</u> - Analoge Systeme - Messverstärker / Verstärkerschaltungen - Analoge Filter - Analog-Digital-Umsetzer - Digital-Analog-Umsetzer <u>Teil 2: Digitale Messtechnik</u> - Analoge und digitale Signale - Zeit/ Frequenz(Fourier-Transformation) - Abtastung und Rekonstruktion - Diskrete Fourier-Transformation, FFT - Spektralanalyse - Korrelationsanalyse - Zeit-Frequenz-Analyse - Laplace- und z-Transformation - Stochastische Signale - Digitale Filterung - Digitale Bildverarbeitung (Einführung)
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Vortrag / Referat oder mündl. Prüfung (30 Min.)

Medienformen:	Beamer (Vorlesungspräsentation), Tafel (Herleitungen, Erläuterungen, Übungen), PDF-Download (Übungen, Vorlesungsskript), Matlab-Übungen
Literatur:	<ul style="list-style-type: none"> <li>- Tietze, U.; Schenk, Ch.: Halbleiterschaltungstechnik, Springer Verlag,</li> <li>- E.O. Brigham, FFT-Anwendungen, Oldenbourg,</li> <li>- K.-D. Kammeyer, K. Kroschel: Digitale Signalverarbeitung, Teubner,</li> <li>- S.D. Stearns, D.R. Hush: Digitale Verarbeitung analoger Signale, Oldenbourg,</li> <li>- A.V. Oppenheim, R.W. Schafer, J.R. Buck: Zeitdiskrete Signalverarbeitung, Pearson.</li> </ul>

Modulbezeichnung:	<b>Computeralgebra 1</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Koepf und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:           2 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	Vorlesung: (2 SWS): 30 h Präsenzzeit Übung: 1 SWS):     15 h Selbststudium Selbststudium:           105 h
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Lineare Algebra und Analysis
Angestrebte Lernergebnisse:	Der / die Studierende - kennt wichtige Strukturen und Methoden der Computeralgebra; - verfügt über grundlegende Problemlösekompetenz; - Kann einfache algorithmische verstehen und eigenständig formulieren; - ist selbstständig in der Lage sich einfache, unbekannte mathematische Sachverhalte und Algorithmen zu erarbeiten; - besitzt die Fähigkeit Computeralgebrasysteme in ersten Algorithmen und bei der Lösung komplexer Aufgaben aus dem Grundbereich Algebra anzuwenden.
Inhalt:	Fähigkeiten von General Purpose-Computeralgebrasystemen; Programmieren in Computeralgebrasystemen; Ganzzahlarithmetik und Arithmetik von Polynomen; Erweiterter euklidischer algorithmus und Anwendungen; Einführung in die Faktorisierung ganzer Zahlen; Einführung in die Faktorisierung von Polynomen.
Studien-/Prüfungsleistungen:	<b>Studienleistung:</b> Erfolgreiche Bearbeitung von Übungsaufgaben. Mindestens 50% der Gesamtpunktzahl müssen erreicht werden. Der Dozent kann für einzelne Lehrveranstaltungen die Bearbeitung der Übungsaufgaben auch ganz oder teilweise durch Kurzreferate oder Hausarbeiten ersetzen. Prüfungsleistung: Klausur oder mündliche Prüfung (120 bzw. 30-45 Min.)
Medienformen:	Die Vorlesung wird unterstützt durch Tafel, Beamer, Moodle, Skripte, Arbeitsblätter und Vorführungen mit Computeralgebrasystemen (Mathematica). Übungen als Programmieraufgaben mit Computeralgebrasystemen. Selbststudium.
Literatur:	Koepf, W.: Computeralgebra.

Modulbezeichnung:	<b><i>Digital Communication Through Band-Limited Channels</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Dahlhaus und Mitarbeiter
Sprache:	Englisch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	3 SWS:        2 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         45 h Präsenzzeit 135 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse der digitalen Kommunikation
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Verfahren der Signalverarbeitung in drahtlosen nachrichtentechnischen Systemen im so genannten Physical Layer (PHY) sowie in der Medienzugriffssteuerung (Medium Access Control, MAC) verstehen und diese für einen Systementwurf einsetzen.
Inhalt:	Carrier and symbol synchronization, signal design for band-limited channels, communication through band-limited linear filter channels, intersymbol interference, adaptive equalization, multicarrier communications
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: mündl. Prüfungen (30 Min.)
Medienformen:	beamer (presentation), board(derivations, explanation), paper (exercices)
Literatur:	J.G. Proakis, Digital Communications, McGraw-Hill, 4 <sup>th</sup> ed., ISBN: 0-07-118183-0.

Modulbezeichnung:	<b><i>Grundlagen der Algebra und Computeralgebra</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dräxler
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	3 SWS:        2 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	150 h:        45 h Präsenzzeit 105 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Lineare Algebra und Analysis
Angestrebte Lernergebnisse:	Der/die Lernende kann: - die elementaren Eigenschaften von Gruppen, Ringen und Körpern zu beschreiben und bewerten, - den euklidischen Algorithmus auf vielfältige Fragen anwenden, - Algorithmen in Körpererweiterungen benennen und durchführen, - Eigenschaften endlicher Körper benennen und bewerten, - einfache mathematische Algorithmen entwerfen, programmieren und ausführen.
Inhalt:	- Grundlegende algebraische Strukturen: Gruppen, Ringe, Moduln, Körper, - Algorithmen zum Rechnen in algebraischen Strukturen - Programmieren von Algorithmen in Computeralgebrasystemen
Studien-/Prüfungsleistungen:	Studienleistung: Erfolgreiche Bearbeitung von Übungsaufgaben (mindest. 50 % der Gesamtpunktzahl); der Dozent kann für einzelne Lehrveranstaltungen die Bearbeitung der Übungsaufgaben auch ganz oder teilweise durch Kurzreferate oder Hausarbeiten ersetzen. Prüfungsleistung, Klausur oder mündliche Prüfung (90 – 150 Min. bzw. 20 - 30Min.)
Medienformen:	Die Vorlesung wird unterstützt durch Tafel, Beamer, Moodle, Skripte, Arbeitsblätter und Vorführungen mit Computeralgebra-Systemen. Übungen als Programmieraufgaben mit Computeralgebra. Selbststudium.
Literatur:	z.B.: Van der Waerden: Algebra I und II, Lang: Algebra, Schultze-Pillott: Elementare Algebra und Zahlentheorie.

Modulbezeichnung:	<b>Grundlagen der Regelungstechnik</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Stursberg und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Kenntnisse zu linearen Differentialgleichungen und linearer Algebra
Angestrebte Lernergebnisse:	Der/die Lernende kann: - grundlegende Eigenschaften dynamischer Systeme erklären, - Ziele der Regelung technischer Prozesse entwickeln, Methoden des Reglerentwurfes für skalare, lineare zeitinvariante Systeme skizzieren, - die Eignung bestimmter Reglertypen für gegebene Systeme und Anforderungen beurteilen und - erhaltene Regelungsergebnisse bewerten.
Inhalt:	Die Lehrveranstaltung behandelt Regelungsmethoden für lineare zeitkontinuierliche Systeme, im Einzelnen: - Erstellung mathematischer Modelle - Verhalten linearer Modelle - Übertragungsfunktionen - Stabilitätsanalyse - Sprungantwort linearer Systeme - Prinzip des Regelkreises - Wurzelortskurvenverfahren - Frequenzkennlinienverfahren - Nyquist-Diagramm - Erweiterte Regelkreisstrukturen - Experimentelle Modellbildung und Modellvereinfachungen - Heuristische Einstellregeln
Studien-/Prüfungsleistungen:	Studienleistungen: Übungsaufgaben Prüfungsleistungen: Klausur oder mündliche Prüfung (90 bzw. 30 Min.)
Medienformen:	Foliensatz zu den wesentlichen Inhalten, Tafelanschrieb, Skript, Übungsaufgaben, Internetseite mit Sammlung sämtlicher relevanter Information und den Dokumenten zur Lehrveranstaltung.
Literatur:	- Skript - H. Unbehauen: Regelungstechnik, Band 1, Vieweg-Verlag, 17. Auflage, 2007, - O. Föllinger: Regelungstechnik, Einführung in die Methoden und ihre Anwendung, Hüthig-Verlag, 10. Auflage, 2008,

	<p>- J. Lunze: Regelungstechnik 1, Springer-Verlag, 7. Auflage, 2008, - R.C. Dorf, R.H. Bishop: Moderne Regelungssysteme, Pearson-Verlag, 1. Auflage 2005.</p>
--	--

Modulbezeichnung:	<b><i>Introduction to Signal Detection and Estimation</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Dahlhaus und Mitarbeiter
Sprache:	Englisch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Basic knowledge of random variables
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Detektions- und Estimationsverfahren verstehen und diese für einen Systementwurf einsetzen.
Inhalt:	Elements of hypothesis testing; mean-squared estimation covering the principle of orthogonality, normal equations, Wiener filters, related efficient numerical methods like Levinson-Durbin recursion, Kalman filters, adaptive filters; classification methods based on linear discriminants, kernel methods, support vector machines; maximum-likelihood parameter estimation, Cramer-Rao bound, EM algorithm
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: mündl. Prüfungen (30 Min.)
Medienformen:	beamer (presentation), board(derivations, explanation), paper (exercices)
Literatur:	H. V. Poor, An Introduction to Signal Detection and Estimation, New York, NY: Springer-Verlag, 2nd ed., 1988. H. van Trees, Detection, Estimation and Modulation Theory, Part I, John Wiley & Sons, 2001. H. van Trees, Detection, Estimation and Modulation Theory, Part II, John Wiley & Sons, 2003. H. van Trees, Detection, Estimation and Modulation Theory, Part III, John Wiley & Sons, 2001. J. M. Mendel, Lessons in Estimation Theory for Signal Processing, Communications, and Control, Prentice-Hall, 1995. C. M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006. D. J. C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003. B. Schölkopf and A. J. Smola, Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond, MIT, 2000.

Modulbezeichnung:	<b><i>Introduction to Information Theory and Coding</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Dahlhaus und Mitarbeiter
Sprache:	Englisch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse der digitalen Kommunikation
Angestrebte Lernergebnisse	<p>Der Student kann</p> <ul style="list-style-type: none"> <li>- grundlegende Zusammenhänge der Informationstheorie anwenden</li> <li>- optimale und suboptimale Verfahren zur Block- und Faltungscodierung und -decodierung entwickeln und anwenden</li> <li>- optimale und suboptimale Verfahren zur Quellencodierung und -decodierung entwickeln und anwenden</li> </ul> <p>Lernergebnisse in Bezug auf die Studiengangsziele:</p> <ul style="list-style-type: none"> <li>- Erwerben von vertieften Kenntnissen in den elektrotechnikspezifischen Grundlagen</li> <li>- Erwerben von erweiterten und angewandten fachspezifischen Grundlagen</li> <li>- Erkennen und Einordnen von komplexen elektrotechnischen und interdisziplinären Aufgabenstellungen</li> <li>- Sicheres Anwenden und Bewerten analytischer Methoden</li> <li>- Selbständiges Entwickeln und Beurteilen von Lösungsmethoden</li> <li>- Tiefgehende und wichtige Erfahrungen in praktischen technischen und ingenieurwissenschaftlichen Tätigkeiten</li> <li>- Arbeiten und Forschen in nationalen und internationalen Kontexten</li> </ul>
Inhalt:	<ul style="list-style-type: none"> <li>- Fundamentals in information theory, mutual information</li> <li>- Typical sequences and Shannon capacity for the discrete memoryless channel</li> <li>- Channel coding: block codes, cyclic block codes, systematic form</li> <li>- Soft and hard decisions and performance; interleaving and code concatenation</li> <li>- Convolutional codes: tree and state diagrams, transfer function, distance properties; the Viterbi algorithm</li> <li>- Source coding: fixed-length and variable-length</li> </ul>

	<p>codes, Huffman coding; the Lempel-Ziv algorithm</p> <ul style="list-style-type: none"> <li>- Coding for analog sources, rate-distortion function; pulse-code modulation; delta-modulation, model-based source coding, linear predictive coding (LPC)</li> </ul>
Studien-/Prüfungsleistungen:	<p>Studienleistungen: keine Prüfungsleistungen: mündl. Prüfung (30 Min.)</p>
Medienformen:	<p>beamer (presentation), board (derivations, explanation), paper (exercices)</p>
Literatur:	<ul style="list-style-type: none"> <li>- T. Cover and J.A. Thomas, Elements of Information Theory, 2<sup>nd</sup> ed., Wiley, ISBN: 978-0-471-241959.</li> <li>- J.G. Proakis, Digital Communications, McGraw-Hill, 4<sup>th</sup> ed., ISBN: 0-07-118183-0.</li> <li>- Papoulis, S.U. Pillai, Probability, Random Variables and Stochastic Processes, McGraw-Hill, 4<sup>th</sup> ed., ISBN: 0071226613.</li> </ul>

Modulbezeichnung:	<b><i>Kryptographie</i></b>
Studiensemester:	Wintersemester 2016/17
Dozent(in):	Prof. Dr. Rück und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Lineare Algebra und Diskrete Strukturen
Angestrebte Lernergebnisse:	Der/die Studierende : - kennt wichtige Strukturen der Kryptographie; - verfügt über grundlegende Problemlösekompetenz - kann mathematische Sachverhalte verstehen, formulieren und in Algorithmen umsetzen; - besitzt die Fähigkeit Fragen der Kryptographie mit Hilfe der Mathematik zu lösen.
Inhalt:	- Klassische Kryptoverfahren - Theorie der Kryptosysteme - Stromchiffren - Blockchiffren - asymmetrische Verfahren - Authentisierung und Datenintegrität
Studien-/Prüfungsleistungen:	Studienleistung: Erfolgreiche Bearbeitung von Übungsaufgaben (mindest. 50% der Gesamtpunktzahl); der Dozent kann für einzelne Lehrveranstaltungen die Bearbeitung der Übungsaufgaben auch ganz oder teilweise durch Kurzreferate oder Hausarbeiten ersetzen. Prüfungsleistung, Klausur oder mündliche Prüfung (90 - 150 Min. bzw. 20-30 Min.)
Medienformen:	Die Vorlesung wird unterstützt durch Tafel und Folien/Beamer/Moodle, Skripte, Selbststudium.
Literatur:	Buchmann: Einführung in die Kryptographie.

Modulbezeichnung:	<b><i>Medium Access Control (MAC) Protocols in Wireless Communications</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Dahlhaus und Mitarbeiter
Sprache:	Englisch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	2 SWS Seminar
Arbeitsaufwand:	90 h:            30 h Präsenzzeit 60 h Selbststudium
Kreditpunkte:	4 CP
Empfohlene Voraussetzungen:	Fundamentals of digital and wireless communication systems
Angestrebte Lernergebnisse:	Der/die Lernende kann - sich Verfahren der Medienzugriffssteuerung (Medium Access Control, MAC) für drahtlose Systeme selbstständig erarbeiten und diese fachlich präsentieren.
Inhalt:	First, we have a short introduction giving an overview over existing MAC protocols in wireless communication systems. Fundamentals schemes will be discussed. These schemes will be presented in greater detail for different wireless communication systems by the participants of the seminar. Topics include distributed vs. Centralized wireless networks, management of collisions, random access, reservation, scheduling, duplexing, fading channels, capacity analysis and important classes of MAC protocols as IEEE 802.11 for wireless local area networks and ad hoc networks.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: mündl. Prüfung (30 Min.)
Medienformen:	Beamer (Präsentation), Tafel (Herleitungen, Erläuterungen), Ausgabe von Publikationen, Präsentationen am Computer.
Literatur:	Material will be handed out during the seminar.

Modulbezeichnung:	<b>Optimierungsverfahren</b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Stursberg und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Lineare Algebra, Analysis
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Typen von Optimierungsproblemen klassifizieren, geeignete mathematische Darstellungen von technischen Optimierungsaufgaben bestimmen, - die Lösung von Optimierungsaufgaben berechnen, die theoretischen Prinzipien der Optimierung durchschauen und algorithmischen Lösungsansätzen zuordnen, - die Optimalität eines Lösungsvorschlags für ein gegebenes Entscheidungsproblem beurteilen und - verschiedene Algorithmen zur mathematischen Optimierung implementieren und anwenden.
Inhalt:	- Einführung in die Optimierung mathematischer Funktionen - Lineare Optimierung - Dualität in konvexer Optimierung - Quadratische Optimierung - Nichtlineare unbeschränkte Optimierung - Nichtlineare Programmierung unter Nebenbedingungen - Diskrete Optimierung - Gemischt-Ganzzahlige Optimierung - Grundprinzipien der stochastischen Optimierung Anwendungsbeispiele
Studien-/Prüfungsleistungen:	Studienleistungen: Übungsaufgaben Prüfungsleistungen: Klausur oder mündl. Prüfung (90 Min. bzw. 30 Min.)
Medienformen:	Foliensatz zu den wesentlichen Inhalten, Tafelanschrieb, Skript, Übungsaufgaben, Internetseite mit Sammlung sämtlicher relevanter Information und den Dokumenten zur Lehrveranstaltung.
Literatur:	- Skript, - J. Nocedal, S.J. Wright: Numerical Optimization, Springer-Verlag, 2006, - R. Fletcher: Practical Methods of Optimization. Wiley, 1987, - S. Boyd, L. Vandenberghe: Convex Optimization. Cambridge Press, 2004,

	<ul style="list-style-type: none"><li>- D. Bertsekas: Nonlinear Programming. Athena Scientific Publ., 1999,</li><li>- G. Nemhauser: Integer and Combinatorial Optimization. Wiley, 1999.</li></ul>
--	--

Modulbezeichnung:	<b><i>Photonische Komponenten und Systeme</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Bangert, Prof. Dr. Hillmer, Prof. Dr. Witzigmann
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundlagenkenntnisse in den Bereichen Optik, elektronische Bauelemente
Angestrebte Lernergebnisse	Der/die Lernende kann: - das Zusammenwirken von photonischen Komponenten in Systemen nachvollziehen, - Problem/Lösungsansätze durch interdisziplinäre Analogien sowie Verständnis von Naturphänomenen formulieren, - theoretische Modellrechnungen aufbereiten, veranschaulichen und mit experimentellen Messwerten vergleichen - grundlegende Prinzipien (Aufbau und Wirkungsweise) photonischer Bauelemente und Systeme sowie Einsatzgrundsätze photonischer Komponenten und System erkennen.
Inhalt:	- Einführung in die Photonik für die Energietechnik, die Mess-Steuer- und Regelungstechnik, die Medizintechnik, die Umweltsystemtechnik, die Sicherheitstechnik, die Informations- und Kommunikationstechnik, die Produktionstechnik und die Kybernetik, - Theoretische Grundlagen: Halbleiter- und Wellenleitermodelle, Fourier-Optik, nichtlineare Optik, - Photonische Komponenten: LED, OLED, Laser (Festkörper, Gas), Photodiode, Solarzellen), - Anwendungen/Systeme: Laser in Produktions- und Medizintechnik, optische Bordnetze, Sensorik/Bio-Chips, Spektroskopie, Beamer, Speichermedien, Beleuchtung.
Studien-/Prüfungsleistung:	Studienleistung: keine Prüfungsleistung: mündl. Prüfung (30 Min.)
Medienformen:	Beamer, Tafel, Overhead-Projektor
Literatur:	- J. Goodman, <i>Introduction to Fourier Optics</i> , 23rd Ed., Roberts & Co., 2005, - R. Menzel, <i>Photonics</i> , Springer, 2007, - E. Hering, <i>Photonik</i> , Springer, 2006, - H. Hillmer, T. Kusserow: <i>Semiconductor Lasers, from Handbook of Lasers</i> , Springer, 2012 (2 <sup>nd</sup> . Ed.), - S. O. Kasap: <i>Optoelectronics and photonics</i> , Prentice Hall, 2001,

	- weitere Literatur wird in der Vorlesung bzw. auf den Homepages der Fachgebiete bekannt gegeben.
--	---

Modulbezeichnung:	<b><i>Simulation of Digital Communication Systems Using MATLAB</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Dozent(in):	Prof. Dr. Dahlhaus und Mitarbeiter
Sprache:	Englisch
Zuordnung zum Curriculum:	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	2 SWS Praktikum
Arbeitsaufwand:	90 h:            30 h Präsenzzeit 60 h Selbststudium
Kreditpunkte:	3 CP
Empfohlene Voraussetzungen:	Knowledge of fundamentals in digital communications
Angestrebte Lernergebnisse:	Der/ die Lernende kann - Verfahren der Signalverarbeitung in nachrichtentechnischen Systemen im so genannten Physical Layer (PHY), insbesondere auch drahtlose Systeme, selbstständig erarbeiten und diese in MATLAB implementieren.
Inhalt:	Introduction to MATLAB and discussion of most important commands, simulation of a simple transmission chain, channel coding (convolutional codes), coding gain, channels with multipath propagation, models of fading channels and performance for binary transmission, Orthogonal Frequency-Division Multiplexing (OFDM), interleaving, implementation of an OFDM modem, Direct Sequence Spread-Spectrum (DSSS) techniques
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: mündl. Prüfung (30 Min.)
Medienformen:	Beamer (presentation), black board (derivations, explanations), paper (exercises).
Literatur:	J.G. Proakis, Digital Communications, New York, NY: McGraw-Hill, 4 <sup>th</sup> ed., 2001.

Modulbezeichnung:	<b><i>Stochastik für Ingenieure</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Alle Dozenten des Institutes Mathematik
Sprache:	Deutsch
Zuordnung zum Curriculum	Vertiefung in Mathematik/Elektrotechnik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse	Der/ die Lernende kann - elementare stochastische Denkweisen beherrschen. - Grundkenntnisse in der stochastischen Modellierung vorweisen und beherrscht die Grundlagen der Schätz- und Testtheorie. - eine statistische Software bedienen und anwenden.
Inhalt:	- Grundkenntnisse in Rund die Erzeugung von Zufallszahlen in R - Wahrscheinlichkeitsraum, Zufallsvariable, Verteilungsfunktion - Diskrete und stetige Verteilungen - Bedingte Wahrscheinlichkeiten, stochastische Unabhängigkeit - Erwartungswert, Varianz, Quantile - Gesetze der großen Zahlen - Kovarianz, Regression - Punktschätzungen - Erwartungstreue, Konsistenz, Maximum-Likelihood-Schätzungen - Tests bei Normalverteilung - Nichtparametrische Tests - Konfidenzintervalle
Studien-/ Prüfungsleistungen:	Studienleistungen: Werden vom Dozenten zu Beginn der Lehrveranstaltung festgelegt. Prüfungsleistung: Schriftliche Prüfung (120-180 min.),
Medienformen:	Tafel und Beamer, Übungen am Computer

<p>Literatur:</p>	<p>Cramer, E. und Kamps, U. (2008). Grundlagen der Wahrscheinlichkeitsrechnung und Statistik. Springer, Berlin. Dalgaard, P. (2002). Introductory Statistics with R. Springer, Berlin. Krenzel, U. (2000). Einführung in die Wahrscheinlichkeitstheorie und Statistik. Vieweg, Braunschweig.</p> <p>DIALEKT-Projekt (2002). Statistik interaktiv. Deskriptive Statistik. Springer, Berlin.</p> <p>Moeschlin, O. (2003). Experimental Stochastics. Springer, Berlin. Sachs, L., Hedderich, J. (2006). Angewandte Statistik. Methodensammlung mit R. Springer, Berlin.</p> <p>R. Schlittgen (2005). Das Statistiklabor. Einführung und Benutzerhandbuch. Springer, Berlin.</p> <p>Verzani, J. (2004). Using R for Introductory Statistics. Chapman &amp; Hall /CRC, London.</p>
-------------------	---

## 2. Module im Bereich Theoretische Informatik

Modulbezeichnung:	<i>Automaten, Spiele, Logiken</i>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Lange und Mitarbeiter
Sprache:	Deutsch/Englisch nach Absprache
Zuordnung zum Curriculum	Theoretische Informatik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Theoretische Informatik: Formale Sprachen und Berechenbarkeit Theoretische Informatik: Logik
Angestrebte Lernergebnisse	Der/die Lernende kann: - vertiefte Kenntnisse im der Grundlagen der Informatik, insbesondere Logik und Theorie formaler Sprachen vorweisen, - theoretische Grundlagen von logischen Spezifikationsprachen und Verifikationstechniken verstehen, - formale Logik in der Modellierung und Beschreibung reaktiver Systeme anwenden, - überdurchschnittliche Abstraktionsfähigkeit vorweisen, - Systemeigenschaften formal spezifizieren und verifizieren, - geeignete Spezifikations- und Verifikationsmechanismen auswählen.
Inhalt:	Die Vorlesung vermittelt die o.g. Kompetenzen im Kontext von Anwendungen, die vor allem aus der Spezifikation und Verifikation reaktiver IT-Systeme, aber auch aus Bereichen wie XML-Anfragesprachen, Computerlinguistik und automatischer Typinferenz stammen. Weiterhin behandelt werden Monadische Logik zweiter Stufe auf endlichen und unendlichen Wörtern; den Satz von Büchi; alternierende Automaten; Baumautomaten; Paritätsspiele; den Satz von Rabin; Monadische Logik zweiter Stufe auf endlichen und unendlichen Bäumen.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Mündliche Prüfung (20 Min.) oder Projektarbeiten
Medienformen:	Die Veranstaltung wird in einer interaktiven Form gehalten, die auf einer selbständigen Vorbereitung seitens der Studierenden basiert.

Literatur:

M. Hofmann, M. Lange: Automatentheorie und Logik,  
eXamen.press, Springer-Verlag, 2010, ISBN 978-3-642-  
18089-7

Modulbezeichnung:	<b><i>Datenbanktheorie</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Lange und Mitarbeiter
Sprache:	Deutsch/Englisch nach Absprache
Zuordnung zum Curriculum	Theoretische Informatik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Datenbanken; Theoretische Informatik - Logik
Angestrebte Lernergebnisse	Der/die Lernende kann: - vertiefte Kenntnisse und sicheren Umgang mit der Ausdrucksstärke von Datenbankanfragesprachen sowie Techniken zum Auswerten und Optimieren von Datenbankanfragen vorweisen
Inhalt:	Die Vorlesung behandelt die Grundlagen von Datenbanken, insbesondere Datenbankanfragesprachen. Sie vermittelt Wissen und Techniken um Fragen um die folgenden Fragen zu beantworten: <ul style="list-style-type: none"> <li>• Wie ausdrucksstark sind Anfragesprachen, sprich welche Information kann man überhaupt mit welchen Anfragesprachen aus einer Datenbank gewinnen?</li> <li>• Was sind die Komplexitäten des Auswerteproblems, sprich wie schwierig ist es überhaupt eine Anfrage auf einer Datenbank auszuwerten? Dies ist wichtig, da ineffiziente Auswerteverfahren auf großen Datenmengen nutzlos sind.</li> <li>• Lassen sich Anfragen optimieren? Dies ist insbesondere für solche Fälle wichtig, in denen eine Anfrage öfters an eine sich möglicherweise mit der Zeit verändernde Datenbank gestellt wird.</li> </ul> <p>Die Vorlesung behandelt dabei drei Arten von Datenbanken. Neben den herkömmlichen relationalen Datenbanken werden auch die Grundlagen für modernere Arten von Datenbanken, nämlich semi-strukturierte Daten und Graphdatenbanken untersucht. Letztere bilden z.B. ein Fundament für das Semantic Web.</p>
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Mündliche Prüfung (20 Min.) oder schriftliche Prüfung
Medienformen:	
Literatur:	z.B. Serge Abiteboul, Richard B. Hull, Victor Vianu: Foundations of Databases. Addison-Wesley, 1995. Weitere Literatur wird in der Vorlesung bekanntgegeben.

Modulbezeichnung:	<b><i>Formale Sprachen und Automaten I</i></b>
Semester:	Wintersemester 2016/2017
Dozent:	Prof. Dr. Friedrich Otto
Sprache:	Deutsch
Zuordnung zum Curriculum:	Theoretische Informatik
Lehrform/SWS	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Inhalt:	<ul style="list-style-type: none"> <li>- endliche Automaten und Transducer</li> <li>- Grammatiken</li> <li>- Pushdown-Automaten</li> <li>- linear beschränkte Automaten</li> <li>- Turingmaschinen</li> <li>- die Chomsky-Hierarchie</li> <li>- Abschlusseigenschaften von Sprachklassen</li> <li>- Entscheidbarkeitsprobleme</li> </ul>
Angestrebte Lernergebnisse :	<p>Der/die Lernende kann</p> <ul style="list-style-type: none"> <li>- das Verhalten von Automaten und Grammatiken beschreiben, vorgegebene Automaten untersuchen und die akzeptierte Sprache bestimmen,</li> <li>- vorgegebene Grammatiken untersuchen und die erzeugte Sprache bestimmen,</li> <li>- Automaten und Grammatiken gemäß der Chomsky-Hierarchie klassifizieren,</li> <li>- Beispielsprachen in die Chomsky-Hierarchie einstufen,</li> <li>- Abschlusseigenschaften herleiten,</li> <li>- Entscheidbarkeit algorithmischer Fragen zu Automaten und Grammatiken folgern.</li> </ul>
Studien-/ Prüfungsleistung:	<p>Studienleistungen: Bearbeitung der wöchentlichen Übungsaufgaben und aktive Teilnahme an den Übungen</p> <p>Prüfungsleistungen: Mündliche Prüfung von 30 Min. Dauer</p>
Medienformen:	Beamer, Tafel, vorlesungsbegleitende Webseite, Selbststudium
Literatur:	<ul style="list-style-type: none"> <li>- Formale Sprachen und Automaten, F. Otto, Vorlesungsskript, Kassel 2015</li> <li>- J. E. Hopcroft, J.D. Ullman; Introduction to Automata Theory, Languages, and Computation; Addison-Wesley, 1979.</li> <li>- G. Rozenberg, A. Salomaa ( Herausgeber ), Handbook of Formal Languages, Vol.1, Springer, Berlin, 1997.</li> <li>- Weitere Literatur wird in der Vorlesung bekannt gegeben.</li> </ul>

Modulbezeichnung:	<b><i>Komplexitätstheorie</i></b>
Semester:	Wintersemester 2016/2017
Dozent:	Prof. Dr. Friedrich Otto
Sprache:	Deutsch
Zuordnung zum Curriculum:	Bereich „Theoretische Informatik“
Lehrform / SWS:	4 SWS Vorlesung : 3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 Stunden : 60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Inhalt:	<ul style="list-style-type: none"> <li>- Turingmaschinen, deterministisch, nicht deterministisch, Orakelmaschinen</li> <li>- Rechenzeit- und Speicherplatzbedarf</li> <li>- Komplexitätsklassen</li> <li>- Hierarchiesätze</li> <li>- Untere Schranken für spezielle Sprachen</li> <li>- Berechenbare Funktionen</li> <li>- Reduzierbarkeit: Polynomialzeit, log. Platz, Turing-Reduzierb.</li> <li>- Vollständigkeit, vollständige Mengen für NP,PSPACE,...</li> <li>- Relativierte Komplexitätsklassen</li> </ul>
Angestrebte Lernergebnisse:	<p>Der/Die Lernende kann</p> <ul style="list-style-type: none"> <li>– das Verhalten der verschiedenen Typen von Turingmaschinen beschreiben,</li> <li>– den Rechenzeit- und den Speicherplatzbedarf von Turingmaschinen abschätzen,</li> <li>– Komplexitätsklassen vergleichen,</li> <li>– die Platz- und Zeitkomplexität von Sprachen analysieren</li> <li>– Sprachen in Komplexitätsklassen einordnen.</li> </ul>
Studienleistung:	Bearbeitung der wöchentlichen Übungsaufgaben und aktive Teilnahme an den Übungen.
Prüfungsleistung:	Mündliche Prüfung von 30 Min. Dauer
Medienformen:	Beamer, Tafel, vorlesungsbegleitende Webseite; Selbststudium
Literatur:	<p>- J.L. Balcazar, J.Diaz, J.Gabarro: Structural Complexity I und II. EATCS Monographs on Theoretical Computer Science, Vol. 11 und 22, Springer, 1988.</p> <p>- M.R. Garey, D.S. Johnson:Computers and Intractability - A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979.</p>

	<p>- J.E. Hopcroft, J.D. Ullmann: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, 1979.</p> <p>- K.R.Reischuk: Komplexitätstheorie. Band I: Grundlagen. Teubner, Stuttgart, 1999.</p>
--	---

Modulbezeichnung:	<b>Reduktionssysteme I</b>
Semester:	Wintersemester 2016/2017
Dozent:	Prof. Dr. Friedrich Otto
Sprache:	Deutsch
Zuordnung zum Curriculum:	Theoretische Informatik
Lehrform / SWS:	4 SWS :        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         60 Stunden Präsenzzeit 120 Stunden Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Inhalt:	<p>Allgemein: Die Veranstaltung behandelt Techniken zum Rechnen in durch Gleichungen definierten Strukturen.</p> <p>Kompetenzen: Das vermittelte Methodenwissen hilft den Studierenden einzuschätzen, ob und gegebenenfalls welche Reduktionstechniken eingesetzt werden können bei der Lösung algorithmischer Probleme in durch Gleichungen definierten Strukturen, wie sie beispielsweise bei der Implementierung funktionaler Sprachen, bei der Programmspezifikation, der automatischen Programmverifikation und der deklarativen Programmierung auftreten.</p> <p>Berufsvorbereitung: Die Veranstaltung bereitet auf den Einsatz in der Softwareentwicklung vor.</p>
Angestrebte Lernergebnisse:	<p>Der/Die Lernende kann:</p> <ul style="list-style-type: none"> <li>-Termersetzungssysteme</li> <li>-Gleichheitstheorien</li> <li>-Satz von Birkhoff</li> <li>-Reduktionsrelationen</li> <li>-Termination und Konfluenz</li> <li>-Knuth-Bendix Vervollständigung</li> </ul>
Studien-/ Prüfungsleistung:	<p>Studienleistungen: Bearbeitung der wöchentlichen Übungsaufgaben und aktive Teilnahme an den Übungen</p> <p>Prüfungsleistungen: Mündliche Prüfung (30 Min.)</p>
Medienformen:	Tafel, Folien, Beamer, vorlesungsbegleitende Webseite, Selbststudium
Literatur:	<p>J. Avenhaus; Reduktionssysteme; Springer, 1995.</p> <p>L. Bachmair; Canonical Equational Proofs; Birkhäuser, 1991.</p> <p>R. Book, F. Otto; String-Rewriting Systems; Springer, New York, 1993.</p> <p>- R. Bündgen; Termersetzungssysteme; Vieweg, 1998.</p>

	<p>- F. Baader, T. Nipkow; Term Rewriting and All That; Cambridge University Press, 1998.</p> <p>- E. Ohlebusch, Advanced Topics in Term Rewriting, Springer, 2002</p>
--	--

Modulbezeichnung:	<b><i>Seminar im Bereich Theoretische Informatik</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Modulverantwortliche/r	Dozenten des Bereichs Theoretische Informatik
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Theoretische Informatik
Lehrform/SWS:	Seminar / 2 SWS
Arbeitsaufwand:	120 h:           ca. 30 h Präsenzzeit ca. 90 h Selbststudium
Kreditpunkte:	4 CP
Empfohlene Voraussetzungen:	Abhängig vom gewählten Seminar
Angestrebte Lernergebnisse:	Der/die Lernende kann: - sich ein Informatikthema selbständig aus der Literatur, inklusive englischsprachiger Originalarbeiten erschließen - im Umgang mit Fachtexten wissenschaftliche Arbeitsmethoden anwenden, z.B. das Hinterfragen von Aussagen, das Bilden eigener Urteile, das Überprüfen von Aussagen, das Hinzuziehen von Sekundärliteratur und das Zusammenfügen von Informationen aus unterschiedlichen Quellen - wissenschaftliche Inhalte für Präsentationen verständlich und strukturiert aufbereiten - in wissenschaftlichen Präsentationen geeignete mündliche und schriftliche Ausdrucksformen einsetzen.
Inhalt:	Abhängig vom konkreten Seminar
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistung: Vortrag (30 – 45 Min.) und Hausarbeit (max. 20 Seiten) oder Vortrag (max. 90 Min.)
Medienformen:	Abhängig vom konkreten Seminar
Literatur:	Abhängig vom konkreten Seminar

Modulbezeichnung:	<b><i>Verifikation eingebetteter Systeme</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Martin Lange
Sprache:	Deutsch
Zuordnung zum Curriculum:	Theoretische Informatik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse theoretische Informatik, eingebettete Systeme, C-Programmierung
Angestrebte Lernergebnisse:	Der/die Lernende kann: - grundlegende Spezifikationsformalismen hinsichtlich Anwendbarkeit beurteilen, - eingesetzte ES Hardware vergleichen, - bestehende Werkzeuge zur Validierung und Verifikation erklären, - neue Werkzeuge und Methoden zu Validierung und Verifikation anwenden.
Inhalt:	Spezifikation mittels StateCharts, Communicating Finite State Machines, Petrinetze, MSCs. Überblick über ES Hardware (Sensoren, Prozessoren, Speicher, Kommunikation), Echtzeitbetriebssysteme. Scheduling (aperiodisch, periodisch, mit Präzedenzen, mit geteilten Ressourcen). Validierung und Verifikation (Echtzeitverifikation, TimedAutomata, UPPAAL, Worst Case Execution Time Analyse). In der gesamten Vorlesung verwenden wir Praxisbeispiele aus dem Smart Grid, Photovoltaik und Smart Home.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Hausarbeit
Medienformen:	Präsentation mit Beamer, Tafel, Papierübungen
Literatur:	- Folien zur Vorlesung - Peter Marwedel, Embedded System Design, Springer, Berlin, 2nd Edition, 2011, - weitere aktuelle Forschungsartikel werden in der Vorlesung bekanntgegeben.

### 3. Module im Bereich Technische Informatik

Modulbezeichnung:	<i>Ausgewählte Kapitel der Automatisierung</i>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Schwarz und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse in der Digitaltechnik
Angestrebte Lernergebnisse:	Der/die Lernende kann: - zustandsbasierte Modelle und Programme nach IEC 61131-3 entwickeln und testen, - technische Netzwerksysteme z.B. basierend auf OPC Client-Server entwerfen, organisieren, testen und analysieren, - den Datenaustausch in heterogenen Systemen konzipieren, organisieren, programmieren und durchführen, - formal Ergebnisse dokumentieren und kritisch bewerten.
Inhalt:	- Strukturierter Entwurf von zustandsbasierten Programmen und Modellen gemäß IEC 61131-3 mit Hilfe der Ablaufsprache, - Strukturierter Entwurf von z.B. OPC Client-Server Architekturen und dezentralen, heterogenen Systemen für den Austausch von Prozessdaten.
Studien-/Prüfungsleistungen:	Studienleistung: 2 Dokumentationen/Berichte Prüfungsleistung: Elektronische Klausur (inkl. Programmieraufgabe) 120-180 Min.
Medienformen:	Beamer, Papier, Tafel, Demonstration und Entwurfsarbeiten am PC
Literatur:	- Günter Wellenreuther, Dieter Zastrow, Automatisieren mit SPS – Theorie und Praxis. Viegweg+Teubner, GMV Fachverlage GmbH, 2011, - L. Litz, Grundlagen der Automatisierungstechnik, Oldenburg Wissenschaftsverlag GmbH 2005, - J. Lunze, Methoden für die Überwachung u. Steuerung kontinuierlicher und ereignisdiskreter Systeme, Oldenburg Wissenschaftsverlag 2003, - W. Mahnke, S.-H. Leitner, M. Damm, OPC United Architecture, Springer Verlag 2009, - F. Iwanitz, J. Lange, OPC : Grundlagen, Implementierung und Anwendung. Heidelberg : Hüthig, 2005

	- weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.
--	--

Modulbezeichnung:	<b><i>Ausgewählte Kapitel der Kommunikationstechnik 2</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. David und Mitarbeiter
Sprache:	Deutsch/Englisch nach Absprache
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der/die Lernende kann: - ausgewählte Themen aus dem Gebiet der Kommunikationstechnik untersuchen und hinterfragen.
Inhalt:	Ausgewählte aktuelle Themen aus dem der Kommunikationstechnik
Studien-/Prüfungsleistungen:	Studienleistungen (b/nb): Referat/Präsentation, Bericht, Anwesenheitspflicht 80% Prüfungsleistungen: mündliche Prüfung oder Klausur (30 bzw. 120 Min.)
Medienformen:	Skript, Folien zum Download, Übungsaufgaben, Software-Vorführungen, eigener Umgang mit realen Produkten/Software.
Literatur:	Die aktuell gültige Übersicht wird in der Einführungsveranstaltung zur Verfügung gestellt.

Modulbezeichnung:	<b><i>Ausgewählte Kapitel der Mikroprozessortechnik</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Dozent(in):	Prof. Dr. Börcsök und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Erfolgreicher Besuch der Vorlesung Mikroprozessortechnik und eingebettete Systeme I und II sowie des Mikroprozessortechnik Labors (Kennziffer. 4331), Rechnerarchitektur.
Angestrebte Lernergebnisse:	Der/die Lernende kann: - die vertieften Kenntnisse moderner Rechner- und Mikroprozessor-Architekturen und Peripherieeinheiten erschließen, - effiziente Programmierung einschätzen und klassifizieren.
Inhalt:	Moderner Rechner- und Mikroprozessor-Architekturen sowie Peripherieeinheiten und deren effiziente Programmierung.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Klausur oder mündliche Prüfung (120 bzw. 40 Min.)
Medienformen:	Beamer, Papier, Tafel, Demonstration, Design- und Entwurfsarbeiten am PC.
Literatur:	- Flik, T., Mikroprozessortechnik, Springer 2001 Hayes, J.P., Computer Architecture and Organisation, McGraw-Hill 1988, - Hennessy, J.L., Computer Architecture, - A quantitative approach , Morgan Kaufmann 2002, - Hwang, K., Advanced Computer Architecture, McGraw Hill 1993, - weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.

Modulbezeichnung:	<b><i>Ausgewählte Kapitel für Programmiersprachen und Techniken nach IEC 61131-3</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Schwarz und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse der Digitaltechnik
Angestrebte Lernergebnisse	Der/die Lernende kann: - Programme, Funktionsblöcke und Funktionen gemäß des internationalen Standards IEC 61131-3 entwickeln und testen, - die Funktionsweise der Sprachelemente erläutern - Programmabläufe mit Hilfe des Standards IEC 61131-3 organisieren, klassifizieren und analysieren, - formal Ergebnisse dokumentieren und kritisch bewerten.
Inhalt:	- Strukturierter Entwurf von Programmen, Funktionsblöcken und Modellen gemäß IEC 61131-3 mit Hilfe von Funktionsbaustein-Sprache und Ablaufsprache, - Einsatz von IEC 61131-3 konformen Sprachelementen, - Einführung in internationale Standards.
Studien-/Prüfungsleistungen:	Studienleistung: 2 Dokumentationen/Berichte Prüfungsleistung: Elektronische Klausur (inkl. Programmieraufgabe) 120-180 Min.
Medienformen:	Beamer, Papier, Tafel, Demonstration und Entwurfsarbeiten am PC
Literatur:	- K.-H. John, M. Tiegelkamp, SPS-Programmierung nach IEC 61131-3, Springer Verlag 2000, - Günter Wellenreuther, Dieter Zastrow, Automatisieren mit SPS – Theorie und Praxis. Viegweg+Teubner, GMV Fachverlage GmbH, 2011, - Karl Pusch, Grundkurs IEC 1131, Vogel Fachbuch 1999, - Heinrich Lepers, SPS Programmierung nach IEC 61131-3 FranzisVerlag GmbH 2005, - weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.

Modulbezeichnung:	<b><i>Communication Technologies I</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. David und Mitarbeiter
Sprache:	Deutsch/Englisch nach Absprache
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Mobile Computing / Introduction to Communication II oder vergleichbar
Angestrebte Lernergebnisse:	Der/die Lernende kann fortgeschrittene und aktuelle Themen auf den Gebieten Data Mining und Context Awareness untersuchen und hinterfragen.
Inhalt:	<ul style="list-style-type: none"> <li>• Fortgeschrittene und aktuelle Themen auf den Gebieten Data Mining und Context Awareness</li> <li>• Anwendung von Algorithmen des maschinellen Lernens auf Applikationen für Context Awareness</li> <li>• Schreiben von wissenschaftlichen Ausarbeitungen und Präsentationen sowie Programmierung von Applikationen für Context Awareness</li> </ul>
Studien-/Prüfungsleistungen:	Studienleistungen (b/nb): Referat/Präsentation, Bericht, Anwesenheitspflicht 80% Prüfungsleistungen: mündliche Prüfung oder Klausur (30 bzw. 120 Min.)
Medienformen:	Skript, Folien zum Download, Übungsaufgaben, Software-Vorführungen, eigener Umgang mit realen Produkten/Software.
Literatur:	<ul style="list-style-type: none"> <li>• "Machine Learning", Peter Flach, Cambridge University Press</li> <li>• "Artificial Intelligence - A Modern Approach", Stuart J Russel and Peter Norvig, Prentice Hall</li> </ul>

Modulbezeichnung:	<b><i>Communication Technologies II</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. David und Mitarbeiter
Sprache:	Deutsch/Englisch nach Absprache
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	ITC2/Rechnernetze
Angestrebte Lernergebnisse:	Der/die Lernende kann: - fortgeschrittene und aktuelle Themen auf dem Gebiet der mobilen Netze und Anwendungen bis hin zu Pervasive Computing untersuchen und hinterfragen.
Inhalt:	Themen wie mobile verteilte System, Middleware, Pervasive Computing, Hausautomatisierung
Studien-/Prüfungsleistungen:	Studienleistungen (b/nb): Referat/Präsentation, Bericht, Anwesenheitspflicht 80% Prüfungsleistungen: mündliche Prüfung oder Klausur (30 bzw.120 Min.)
Medienformen:	Skript, Folien zum Download, Übungsaufgaben, Software-Vorfürhungen, eigener Umgang mit realen Produkten/Software.
Literatur:	Die aktuell gültige Übersicht wird in der Einführungsveranstaltung zur Verfügung gestellt:

Modulbezeichnung:	<b><i>Computational Intelligence in der Automatisierung</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr.-Ing. Andreas Kroll,
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS            3 SWS Vorleseun 1 SWS Übung
Arbeitsaufwand:	180 h            60 h Präsenzzeit: 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Einführung in Regelungstechnik
Angestrebte Lernergebnisse:	<p>Der/die Lernende kann</p> <ul style="list-style-type: none"> <li>• grundlegende Begriffe, Konzepte und Methoden der Computational Intelligence (CI) mit ihren drei Teilgebieten Fuzzy-Logik, künstliche Neuronale Netze und Evolutionäre Algorithmen verstehen</li> <li>• einfache CI-Anwendungen selbstständig und systematisch zu erstellen.</li> <li>• Des Weiteren erwerben Studierende eine ausreichende Kompetenz, um die Eignung von CI-Methoden zur Lösung einer technischen Aufgabe abschätzen zu können. Sie können die entsprechende technisch- wissenschaftliche Literatur lesen.</li> </ul>
Inhalt:	<ul style="list-style-type: none"> <li>• Was bedeutet Computational Intelligence und was ist das besondere an ihr?</li> <li>• Fuzzy-Logik und Fuzzy-Systeme <ul style="list-style-type: none"> <li>○ Grundlegende Begriffe und Konzepte</li> <li>○ Fuzzy Control</li> <li>○ Fuzzy-Modellierung, Fuzzy-Identifikation</li> <li>○ Fuzzy- Klassifikation</li> <li>○ Anwendungsbeispiele</li> </ul> </li> <li>• Künstliche Neuronale Netze <ul style="list-style-type: none"> <li>○ Grundlegende Begriffe und Konzepte</li> <li>○ Netzwerke vom MLP-, RBF- und SOM-Typ</li> <li>○ Anwendungsbeispiele</li> </ul> </li> <li>• Evolutionäre Algorithmen <ul style="list-style-type: none"> <li>○ Grundlegende Konzepte</li> <li>○ Genetische Algorithmen</li> <li>○ Evolutionäre Strategien</li> <li>○ Anwendungsbeispiele</li> </ul> </li> <li>• Ausblick: Schwarmintelligenz &amp; künstliche Immunsysteme</li> </ul>
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: schriftliche (120 min.) oder mündliche (30 min.)
Medienformen:	<ul style="list-style-type: none"> <li>- Ausdruckbares Skript (PDF)</li> <li>- Beamer</li> <li>- Web-Portal zum Kurs mit Skript zum Herunterladen und Zusatzinformationen</li> </ul>

	- Tafel
Literatur:	<ul style="list-style-type: none"><li>• Basisliteratur: A. P. Engelbrecht, Computational Intelligence-an introduction. Chichester: Wiley, 2002. ISBN: 0-470-84870-7</li><li>• Vertiefende Literatur spezifisch zu den einzelnen</li><li>• Themenabschnitten</li><li>• Skript</li></ul>

Modulbezeichnung:	<b><i>Computerarithmetik/Computer Arithmetic</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Zipf, Dr. –Ing. Martin Kumm
Sprache:	Deutsch, Englisch nach Absprache möglich
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Digitale Logik, wünschenswert aber keine notwendige Voraussetzung: Rechnerarchitektu
Angestrebte Lernergebnisse:	<p>Der/die Lernende kann,</p> <ul style="list-style-type: none"> <li>- den Aufbau arithmetischer Einheiten moderner Computer beurteilen,</li> <li>- unterschiedliche Darstellungen von Zahlen auf Computern anwenden,</li> <li>- arithmetische Einheiten für Grundrechenarten sowie elementarer Funktionen entwerfen.</li> </ul> <p>The student is able to</p> <ul style="list-style-type: none"> <li>- evaluate the construction of arithmetic units in modern computers,</li> <li>- apply different number representations used in computers,</li> <li>- design arithmetic units for basic arithmetic and elementary functions.</li> </ul>
Inhalt:	<ul style="list-style-type: none"> <li>- Zahlendarstellungen (Festkomma-/Gleitkommaformat, Darstellung negativer Zahlen, alternative Zahlensysteme)</li> <li>- Addition/Subtraktion (Ripple-Carry Addierer, Carry-Lookahead Addierer, Parallel Prefix Adder)</li> <li>- Compressor Trees (Wallace Tree, Dadda Tree)</li> <li>- Multiplikation (Baugh-Wooley- und Booth-Multiplizierer, Higher Radix Multiplizierer)</li> <li>- Division (Restoring/Non-restoring Division, SRT Division)</li> <li>- Funktions-Approximation (Normalisierung und Bereichsreduktion, Polynom-, Rational- und Spline-Approximation, CORDIC Algorithmus, Multipartite Table Methode)</li> <li>- Gleitkomma-Arithmetik (Addition/Subtraktion, Multiplikation, Division)</li> <li>- Besonderheiten auf FPGAs</li> </ul> <ul style="list-style-type: none"> <li>- Number representation (fixed point numbers, floating point numbers, negative numbers, alternative number systems)</li> <li>- Addition/subtraction (ripple-carry adder, carry lookahead adder, parallel prefix adder)</li> </ul>

	<ul style="list-style-type: none"> <li>- Compressor Trees (Wallace tree, Dadda tree)</li> <li>- Multiplication (Baugh-Wooley multiplier, Booth multiplier, higher radix multipliers)</li> <li>- Division (Restoring/non-restoring division, SRT division)</li> <li>- Function approximation (normalization, range reduction, polynomial approximations, rational approximations and splines, CORDIC algorithm, multipartite table methods)</li> <li>- Floating point arithmetic (Addition/subtraktion, multiplication, division)</li> <li>- Particularities for FPGAs</li> </ul>
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Mündl. Prüfung (etwa 40 Min.)
Medienformen:	Folien/Beamer, Tafel
Literatur:	<p>Parhami, B. (2009). Computer Arithmetic - Algorithms and Hardware Designs. Oxford University Press.</p> <p>Muller, J.-M. (2006). Elementary Functions (2nd ed.). Boston, MA: Springer Science &amp; Business Media.</p>

Modulbezeichnung:	<b><i>Einführung in die Virtuelle Realität</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Wloka
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	C++ , Computergraphik
Angestrebte Lernergebnisse	Der/die Lernende kann - vorgegebene bzw. bekannte Techniken bzw. Algorithmen aus der virtuellen Realität erklären, - Virtual Reality Systeme entwickeln.
Inhalt:	- Grundkonzepte 3D / Stereo, - VR-Hardware, Fishtank bis CAVE, - VR-Eingabesysteme, - Softwarearchitekturen und –systeme, - Anwendungsgebiete, - VR-Juggler.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Hausarbeit
Medienformen:	Multimedia-Präsentationen, Übungen mittels E-Learning (moodle), elektronische Kommunikationsplattform (moodle)
Literatur:	Skript, Bücher werden in Vorlesung bekannt gegeben Internetquellen, Programmtutorials.

Modulbezeichnung:	<b><i>Java Code-Camp Context Awareness 2</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. David und Mitarbeiter
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	Blockveranstaltung max. 10 Teilnehmer 180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Java bzw. Android/IOS-Kenntnisse
Angestrebte Lernergebnisse	<ul style="list-style-type: none"> <li>• Anwenden von Programmierkenntnissen in Java bzw. Android/IOS im kontextsensitiver bzw. mobiler Anwendungen</li> <li>• Verstehen und analysieren objektorientierte Konzepte</li> <li>• Entwerfen der Anwendungsarchitektur</li> <li>• Planen des Entwicklungsablaufs</li> <li>• Bewerten von Sensordaten</li> </ul>
Inhalt:	Die Veranstaltung beinhaltet das Programmierung von aktueller Sensorik und Aktorik sowie die Programmierung mit Java bzw. Android/IOS
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistung: Ausarbeitung, Dokumentation und Präsentation (30 Min.)
Medienformen:	Powerpoint
Literatur:	Understanding Object-Oriented Programming with Java von Timothy A. Budd (ISBN-10: 0201308819)

Modulbezeichnung:	<b>Prozessrechner</b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Dozent(in):	Prof. Dr. Börcsök und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS: 4 SWS Vorlesung
Arbeitsaufwand:	180h: 60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Kenntnisse der Mikroprozessor- und Regelungstechnik
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Aufbau und Wirkungsweise von Prozessrechnersystemen klassifizieren, - die Hard- und Softwarekomponenten einstufen und bewerten, sowie die Steuerungsmöglichkeiten mittels Prozessrechner ableiten. - die Möglichkeiten zur Modellierung der zu steuernden oder zu regelnden Prozesse und deren mathematische Beschreibungen bewerten und einstufen.
Inhalt:	Struktur von Prozessen, Mathematische Modellbeschreibungen, Aufbau von Prozessrechner- und Automatisierungssystemen, Aufbau und Wirkungsweise von Peripherieeinheiten, Echtzeiteigenschaften Programmierung und Werkzeugauswahl, Vorstellung marktüblicher Systeme und Werkzeuge mit Bezug auf die Anwendung.
Studien-/Prüfungsleistungen:	Studienleistungen: Hausarbeit, Referat/Präsentation Prüfungsleistungen: Klausur oder mündliche Prüfung (120 bzw. 40 Min.)
Medienformen:	PPT-Folien, Tafel, Demonstration, Arbeiten am PC
Literatur:	- Heidepriem, Prozessinformatik 1, Oldenburg 2000, - Heidepriem, Prozessinformatik 2, Oldenburg 2001, - Lauber, R., Prozessautomatisierung, Springer 1989, - Färber, G. Prozessrechentchnik, Springer 1994, - Börcsök, J. Prozessrechner und Automation, Heise 1999, - weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.

Modulbezeichnung:	<b>Rechnergestützter Entwurf mikroelektronischer Schaltungen</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Zipf und Mitarbeiter
Sprache:	Deutsch, Englisch nach Absprache möglich
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	3 SWS:        2 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         45 h Präsenzzeit 135 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Kenntnisse in diskreter Mathematik
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Ablauf und Ziele des physikalischen Entwurfs skizzieren, - vorgegebene bzw. bekannte Algorithmen erklären, - Teilalgorithmen zu einem Gesamtablauf kombinieren, - Implementierungen gegebener Algorithmen vergleichen, - Implementierungen von Algorithmen entwickeln, - Platzierungs- und Verdrahtungsergebnisse qualitativ beurteilen, - Simulationsverfahren erklären und klassifizieren.
Inhalt:	Aufbauend auf den theoretischen Grundlagen werden, jeweils dem Entwurfsablauf folgend, die Methoden und Algorithmen diskutiert, die die Basis für aktuelle industrielle CAD-Systeme für den Chipentwurf bilden. Damit wird ein tiefgehendes Verständnis für deren Funktionsweise gefördert und ein zielgerichteter Einsatz dieser Tools ermöglicht. Behandelt werden u.a. Optimierungsmethoden, Algorithmen im physikalischen Entwurf (Partitionierung, Platzierung, Verdrahtung) sowie Simulationsalgorithmen.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Klausur oder mündl. Prüfung (90 bzw. 40 Min.)
Medienformen:	Folien/Beamer, Tafel
Literatur:	- Sabih H. Gerez: Algorithms for VLSI Design Automation, John Wiley & Sons, 1. Auflage, 1998 - Naveed A. Sherwani: Algorithms for VLSI Physical Design Automation, Springer Verlag; 3. Auflage. 1999 - Michael J. S. Smith: Application-Specific Integrated Circuits, Addison-Wesley Longman, 1997 - Jens Lienig: Layoutsynthese elektronischer Schaltungen, Springer Verlag, 1. Auflage, 2006 - Reinhard Diestel: Graphentheorie, Springer, Berlin; 3. Auflage, 2006 - weitere Literatur wird in der Vorlesung bzw. auf der Homepage des Fachgebiets bekannt gegeben.

Modulbezeichnung:	<b>Rekonfigurierbare Strukturen</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Zipf und Mitarbeiter
Sprache:	Deutsch, Englisch nach Absprache möglich
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Kenntnisse im Entwurf digitaler Schaltungen (Bachelor-Level), Algorithmen & Datenstrukturen (Bachelor-Level) sowie in diskreter Mathematik (Grundkenntnisse).
Angestrebte Lernergebnisse	Der/die Lernende kann: - den prinzipiellen Aufbau von FPGAs skizzieren, - Methoden der Platzierung und Verdrahtung sowie deren Zusammenhang erklären, - Quantitative Architekturentscheidungen begründen, - verschiedene Architekturmodelle und Rekonfigurationsverfahren beschreiben und bewerten, - eigene Architekturvorschläge entwickeln, - Verfahren der dynamischen Rekonfiguration erklären, - Einsatzmöglichkeiten von FPGAs einschätzen.
Inhalt:	Funktionsweise und innerer Aufbau von FPGAs und anderen rekonfigurierbaren bzw. strukturell programmierbaren Schaltungen. Behandelt werden zunächst FPGAs und die Grundlagen der zur ihrer Programmierung verwendeten Software-Tools sowie deren Optimierungsziele und -methoden. Darauf aufbauend werden weitere grob- und eingranulare Architekturen und Techniken der dynamischen Rekonfiguration besprochen. Darüber hinaus werden die Grundlagen gelegt, selbst rekonfigurierbare Architekturelemente und Rekonfigurationskonzepte in Chip- und Schaltungsentwurfsprojekten einzubringen, wie sie in vielen Firmen inzwischen benötigt werden.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: mündl. Prüfung (etwa 40 Min.) oder Hausarbeit mit Präsentation
Medienformen:	Folien/Beamer, Tafel, Rechnerübung
Literatur:	- Scott Hauck, Andre DeHon (Hrsg.): Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation, Morgan Kaufmann Series in Systems on Silicon, Academic Press, 2007, - Vaughn Betz, Alexander Marquardt, Jonathan Rose: Architecture and CAD for Deep-Submicron FPGAs, Springer Verlag, 1999,

- |  |  |
|--|--|
|  | <ul style="list-style-type: none"><li>- Dimitrios Soudris, Stamatis Vassiliadis (Hrsg.): Fine- and Coarse-Grain Reconfigurable Computing, Springer-Verlag, 2007,</li><li>- Ramachandran Vaidyanathan, Jerry Trahan: Dynamic Reconfiguration: Architectures and Algorithms (Series in Computer Science), Springer Netherlands, 2003,</li><li>- weitere Literatur wird in der Vorlesung bzw. auf der Homepage des Fachgebiets bekannt gegeben.</li></ul> |
|--|--|

Modulbezeichnung:	<b><i>Risikobewertung von Rechnerarchitekturen 2</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Dozent(in):	Prof. Börcsök und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:      2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Programmierkenntnisse, Grundlagen der Informatik, Digitaltechnik, Mikroprozessoren oder Rechnerarchitektur, Regelungstechnik, Mathematik
Angestrebte Lernergebnisse:	Der/die Lernende kann: - das Risiko von unterschiedlichen Rechnerarchitekturen bestimmen, - Risikopotentialen in Hard- und Softwarekomponenten, Grundlagen der mathematische Modelle und Beschreibungen bestimmen.
	Die Vorlesung beschäftigt sich mit Risikoberechnung, Wahrscheinlichkeitstheorie, Struktur von Rechnerarchitekturen, Mathematische Modellbeschreibungen, Berechnungen der Architekturmodelle.
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistungen: Klausur (60-180 Min.) oder mündl. Prüfung (20 – 40 Min.)
Medienformen:	Folien, Script, Vortrag
Literatur:	Skript, wird zu Veranstaltungsbeginn ausgegeben.  Weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.

Modulbezeichnung:	<b>Schaltungsentwurf mit HDLs</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Zipf und Mitarbeiter
Sprache:	Deutsch, Englisch nach Absprache möglich
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Vorlesung Digitale Logik oder äquivalente Kenntnisse
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Grundelemente einer Hardwarebeschreibungssprache benennen, - die Funktionsweise der Sprachelemente erläutern, - in einer HDL beschriebene Schaltungen interpretieren, - Beschreibungen von Standardschaltungen in einer HDL entwerfen, - mit Synthesesoftware Entwürfe implementieren.
Inhalt:	Syntax und Semantik von VHDL, Modellierungsansätze, Beschreibung von Standardschaltungen (Schaltnetze, Zustandsautomaten, Datenpfadfunktionalität), Systementwurf, Synthese von konkreten Schaltungen mit kommerzieller CAD-Software
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Klausur oder mündl. Prüfung (90 bzw. 40 Min.)
Medienformen:	Folien/Beamer, Tafel, Rechnerübungen
Literatur:	- Peter Ashenden: The Designer's Guide to VHDL, Morgan Kaufmann; 3. Auflage, 2006, - Paul Molitor, Jörg Ritter: VHDL: Eine Einführung, Pearson Studium, 2004, - Jürgen Reichardt, Bernd Schwarz: VHDL-Synthese: Entwurf digitaler Schaltungen und Systeme, Oldenbourg, 5. Auflage, 2009, - Frank Kesel, Ruben Bartholomä: Entwurf von digitalen Schaltungen und Systemen mit HDLs und FPGAs: Einführung mit VHDL und SystemC, Oldenbourg; 2. Auflage, 2009, - weitere Literatur wird in der Vorlesung bzw. auf der Homepage des Fachgebiets bekannt gegeben.

Modulbezeichnung:	<b><i>Seminar im Bereich Technische Informatik</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Modulverantwortliche/r	Dozenten des Bereichs Technische Informatik
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	Seminar / 2 SWS
Arbeitsaufwand:	120 h:           ca. 30 h Präsenzzeit ca. 90 h Selbststudium
Kreditpunkte:	4 CP
Empfohlene Voraussetzungen:	Abhängig vom gewählten Seminar
Angestrebte Lernergebnisse:	Der/die Lernende kann: - sich ein Informatikthema selbständig aus der Literatur, inklusive englischsprachiger Originalarbeiten erschließen - im Umgang mit Fachtexten wissenschaftliche Arbeitsmethoden anwenden, z.B. das Hinterfragen von Aussagen, das Bilden eigener Urteile, das Überprüfen von Aussagen, das Hinzuziehen von Sekundärliteratur und das Zusammenfügen von Informationen aus unterschiedlichen Quellen - wissenschaftliche Inhalte für Präsentationen verständlich und strukturiert aufbereiten - in wissenschaftlichen Präsentationen geeignete mündliche und schriftliche Ausdrucksformen einsetzen.
Inhalt:	Abhängig vom konkreten Seminar
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistung: Vortrag (30 – 45 Min.) und Hausarbeit (max. 20 Seiten) oder Vortrag (max. 90 Min.)
Medienformen:	Abhängig vom konkreten Seminar
Literatur:	Abhängig vom konkreten Seminar

Modulbezeichnung:	<b>Signal- und Bildverarbeitung</b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr.-Ing. Andreas Kroll, Dr.-Ing. Werner Baetz
Sprache:	Deutsch,
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 1 SWS Übung 1 SWS Praktikum
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der/die Lernende kann: - die grundlegenden Funktionen der Signal- und Bildverarbeitung verstehen und anwenden. Er/Sie kann deterministische und stochastische Signale im Zeit- bzw. Orts- und Spektral beschreiben und versteht die Zusammenhänge zur digitalen Analyse und Verbesserung von Zeit- und Bildsignalen. - Ferner kennt Er/Sie Methoden zur Störunterdrückung und Identifikation gestörter linearer Systeme.
Inhalt:	- Definition von Zeit- und Bildsignalen und ihre analytischen Beschreibungsformen (z. B. deterministische und stochastische Signale, Energie- und Leistungssignale) - Strukturen und Elemente signalverarbeitender Systeme - Methoden der Signalverarbeitung im Zeit- und Ortsbereich, (z. B. Zeitdiskretisierung, Digitalisierung, z-Transformation, FFT, Filterung, Mittelung, Korrelationsfunktionen, Lock-In-Verfahren, Modulation, Demodulation, etc.) - Methoden der Signalverarbeitung im Spektral (auch Ortsfrequenzbereich), (z. B. Fensterung, Aliasing, Diskrete- Fouriertransformation, Amplituden-, Phasen- und Leistungsdichtespektren, Kohärenzfunktion, - Rauschen, Filterung, Multi-Sensor-Datenfusion - Anwendung von Werkzeugen zur digitalen Signalverarbeitung anhand von Rechnersimulationen zur Vertiefung der Methodenkenntnisse.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: schriftliche (120 min.) oder mündliche (30 min.)
Medienformen:	- Ausdruckbares Skript (PDF) - Beamer - Web-Portal zum Kurs mit Skript zum Herunterladen und Zusatzinformationen - Tafel - PC-Pool für praktische Übungen und Anwendung der Signalverarbeitungsmethoden

<p>Literatur:</p>	<ul style="list-style-type: none"> <li>- Meffert, B., Hochmuth, O.: Werkzeuge der Signalverarbeitung, Pearson Studium, 2004, ISBN 3-8273-7065-5</li> <li>- Von Grünigen, D. Ch.: Digitale Signalverarbeitung, Fachbuchverlag Leipzig Hanser Verlag München, 2002 ISBN 3-446-21976-5</li> <li>- Ohm, J.-R., Lüke, H. D.: Signalübertragung – Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme, Springer 2006, ISBN 3540222073</li> <li>- Meyer, M: Signalverarbeitung; Analoge und digitale Signale, Systeme und Filter, Vieweg+Teubner Verlag, 2006, ISBN 3834802433</li> <li>- Tönnies, K. D.: Grundlagen der Bildverarbeitung, Pearson Studium, 2005, ISBN 3-8273-7155-4</li> </ul>
-------------------	---

Modulbezeichnung:	<b><i>Synthese und Optimierung mikroelektronischer Systeme</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Zipf und Mitarbeiter
Sprache:	Deutsch, Englisch nach Absprache möglich
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	3 SWS:        2 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180 h:         45 h Präsenzzeit 135 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Kenntnisse in diskreter Mathematik und im Entwurf digitaler Schaltungen (Bachelor-Level)
Angestrebte Lernergebnisse:	Der/die Lernende kann: - den Ablauf und die Ziele der High-Level Synthese skizzieren, - vorgegebene bzw. bekannte Algorithmen erklären, - Implementierungen gegebener Algorithmen vergleichen, - Erweiterungen für vorhandene Algorithmen entwickeln, - Syntheseergebnisse qualitativ beurteilen.
Inhalt:	Einführung in die High-Level-Synthese (HLS) und die dort eingesetzten Algorithmen. Als Teil des Systementwurfs führt die HLS zu Systemimplementierungen. Die Vorlesung bietet eine Übersicht über den allgemeinen Systementwurfsablauf sowie die in CAD-Systemen eingesetzten Optimierungsansätze und konkreten Optimierungsalgorithmen, wie sie derzeitigen Softwaresystemen im industriellen Einsatz zugrunde liegen. Detailliert behandelt werden Algorithmen und Verfahren im HW/SW Codesign, in der High-Level-Synthese, der Register-Transfer-Synthese sowie bei der Register-Transfer-Optimierung.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Mündl. Prüfung (ca. 40 Min.) oder Hausarbeit mit Präsentation
Medienformen:	Folien/Beamer, Tafel
Literatur:	- Giovanni DeMicheli: Synthesis and Optimization of Digital Circuits, McGraw-Hill Publ. Comp., 1994, - Petra Michel, Peter Duzy, Ulrich Lauther (Hrsg.): The Synthesis Approach to Digital System Design, Kluwer International Series in Engineering & Computer Science, 2. Auflage, 1992, - Volker Turau: Algorithmische Graphentheorie, Oldenbourg, 3. Auflage, 2009, - weitere Literatur wird in der Vorlesung bzw. auf der Homepage des Fachgebiets bekannt gegeben.

Modulbezeichnung:	<b><i>Technik im Bereich neuer Medien</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Wloka und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	C++ und Computergraphik
Angestrebte Lernergebnisse:	Der/die Lernende kann: - vorgegebene bzw. bekannte Techniken und Algorithmen aus dem Technik neuer Medien erklären, - Avatare selbstständig konzipieren und entwickeln.
Inhalt:	- Animationskonzepte - Modellierung von Avataren für Animationen - Physikalische Simulation - Animation von Avataren
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistung: Hausarbeit (30 % semesterbegleitende Projekte und 70 % Endprojekt) Weitere Informationen siehe moodle
Medienformen:	Multimedia-Präsentationen, Übungen mittels E-Learning (moodle), elektronische Kommunikationsplattform (moodle)
Literatur:	Skript, Bücher werden in Vorlesung bekannt gegeben Internetquellen, Programmtutorials

Modulbezeichnung:	<b><i>Theorie sicherheitsgerichteter Rechnersysteme</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Börcsök und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS: 4 SWS Vorlesung
Arbeitsaufwand:	180h: 60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der/die Lernende kann: - Modelldefinitionen von sicherheitsgerichteten Rechnerarchitekturen bewerten und beurteilen, - Zuverlässigkeits- und Sicherheitsparameter für unterschiedliche Architekturmodelle ableiten und analysieren.
Inhalt:	Mathematische Modelle von Komponenten und Systemen, Funktionsblock- und Markov-Analyse und Berechnung gegebener Architekturmodelle, Modellbeschreibungen, Test-, Prüfverfahren,
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Klausur oder mündliche Prüfung (120 bzw. 40 Min)
Medienformen:	Beamer, Papier, Tafel, Demonstration am PC
Literatur:	- Barlow, R. E., Engineering Reliability, ASA.SIAM 1998, - Bitter, P., Technische Zuverlässigkeit, Springer 1977 - Leitch, R. D., Reliability Analysis for Engineers, Oxford Science Publication 1995, - Börcsök, J. Electronic Safety Systems, Hüthig 2004 Neumann, P. Computer Related Risk, Addison Wesley 1995, - Goble, W., Evaluation Control Systems Reliability, ISA 1992, - Skript (wird zu Veranstaltungsbeginn ausgegeben) - weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.

Modulbezeichnung:	<b><i>Zuverlässigkeitstheorie für Rechnersysteme</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Börcsök und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Technische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der/die Lernende kann: - unterschiedliche Rechnerarchitekturen bewerten und beurteilen, - Zuverlässigkeitsparameter bestimmen, - die Klassifizierung gegebener Architekturmodelle ableiten.
Inhalt:	Wahrscheinlichkeits-, Verfügbarkeits- und Sicherheitsbetrachtung von Rechnersystemen, mathematische Modellbeschreibungen unterschiedlicher Rechnersysteme. Funktionsblockanalyse, Markovmodell, etc. Test-, Prüfverfahren.
Studien-/Prüfungsleistungen:	Studienleistungen: Hausarbeit, Referat/Präsentation Prüfungsleistungen : Klausur oder mündliche Prüfung (120 bzw. 40 Min)
Medienformen:	Beamer, Tafel, Papier, Demonstration an PC und Modellen
Literatur:	- Barlow, R. E., Engineering Reliability, ASA.SIAM 1998, - Bitter, P., Technische Zuverlässigkeit, Springer 1977, - Leitch, R. D., Reliability Analysis for Engineers, Oxford Science Publication 1995, - Börcsök, J. Electronic Safety Systems, Hüthig 2004, - Skript (wird zu Veranstaltungsbeginn ausgegeben), - weitere Literatur wird in der Lehrveranstaltung bekannt gegeben.

## 4. Module im Bereich Praktische Informatik

Modulbezeichnung:	<i>Graph &amp; Model Driven Engineering</i>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Zündorf und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180 h:        60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der Student kann: - grafisch vorgegebene Modelle und Modelltransformationen formalisieren und die Anwendung einer Modelltransformation auf ein Modell formal beschreiben, - formal beschriebene vorgegebene Modelle und Modelltransformationen grafisch darstellen und die Anwendung einer formal definierten Modelltransformation auf ein formal definiertes Modell grafisch darstellen, - Modelle und Modelltransformationen mit Hilfe verschiedener Rahmenwerke implementieren und einen Modelltransformations-Interpreter entwerfen und implementieren.
Inhalt:	Theorie und Praxis von graphartigen Modellen und Modelltransformationen. Graphartige Modelle und Modelltransformationen werden mit mengentheoretischen Mitteln formalisiert. Darauf aufbauend werden Modelle und Modelltransformationen mit verschiedenen Modelleriungs-Frameworks praktisch realisiert. Dies umfasst auch die Implementierung eines Interpreters für Modelltransformationen.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Klausur oder mündliche Prüfung über Formalisierung und Deformalisierung von Modellen und Modelltransformationen und ihre Ausführung. Projekt: Implementierung von Modellen, Modelltransformationen und Interpreter mit Hilfe eines geeigneten Rahmenwerks.
Medienformen:	Tafel, Folien, Programmdemonstrationen, Vorlesungsbegleitende Webseite, Selbststudium
Literatur:	Wird in der Vorlesung bekannt gegeben

Modulbezeichnung:	<b><i>Home Automation</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Ulrich Norbistrath und Prof. Albert Zündorf
Sprache:	Deutsch/Englisch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180 h:         60 h Präsenzzeit 120 h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	C
Angestrebte Lernergebnisse:	Die Studierenden können umfangreiche Heimautomatisierungsprojekte durchführen. Die Studierenden können dabei verschiedene Standards und Techniken für Microcontroller, Sensorik und Aktorik einsetzen. Die Studierenden können verschiedene Vernetzungstechniken in der Heimautomatisierung verwenden.
Inhalt:	Techniken für die Heimautomatisierung: <ul style="list-style-type: none"> <li>- Microcontroller Boards</li> <li>- Sensorik und Aktorik</li> <li>- Vernetzung</li> <li>- Automatisierung</li> <li>- Sicherheitsaspekte</li> </ul>
Studien-/Prüfungsleistungen:	Studienleistungen: Hausaufgaben Prüfungsleistungen: Projektaufgabe
Medienformen:	Folien, Web-Sites
Literatur:	Wird in der Vorlesung bekannt gegeben

Modulbezeichnung:	<b><i>Lernen in kollaborativen Multi-Agenten Systemen / Learning in Collaborative Multi-Agent Systems</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Geihs und Mitarbeiter
Sprache:	Englisch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	- Computer science basics / Grundlagen der Informatik - Introduction to Programming / Einführung der Programmierung
Angestrebte Lernergebnisse:	The understanding of collaborative distributed systems esp. Multi-Agent Systems (MAS), whose intelligence obtained after performing a specific Machine Learning method, e.g., decentralized market control, a team of robotic soccer.  Verständnis der kollaborativen verteilten Systeme, insbesondere Multi-Agenten Systeme (MAS), deren Intelligenz durch maschinelle Lernverfahren gestaltet wird, z.B. dezentralisierte Marktbeherrschung, ein Team von Fußballrobotern.
Inhalt:	<ul style="list-style-type: none"> <li>• Agent model and Self-X Properties</li> <li>• Collaboration and competition in Multi-agent Systems</li> <li>• Nature-inspired algorithms</li> <li>• Machine learning, esp. reinforcement learning</li> <li>• Real application examples: Robotic soccer team and more</li> </ul>
Studien-/Prüfungsleistungen:	Studienleistungen: Exercises/ Übungen Prüfungsleistungen: oral exam / mündliche Prüfung (25 Minuten) or written exam / schriftliche Prüfung (120 Minuten)
Medienformen:	Media: Slides, Board, Exercise Sheets, Course Associated Web Page ( <a href="http://www.vs.uni-kassel.de">www.vs.uni-kassel.de</a> ), Utilisation of NetLogo (free Multiagenten-Simulator), <a href="http://ccl.northwestern.edu/netlogo/index.shtml">http://ccl.northwestern.edu/netlogo/index.shtml</a>  Medienformen: Folien, Tafel, Übungsblätter, vorlesungsbegleitende Web Page ( <a href="http://www.vs.uni-kassel.de">www.vs.uni-kassel.de</a> ), Verwendung von NetLogo (kostenloser Multiagenten-Simulator), <a href="http://ccl.northwestern.edu/netlogo/index.shtml">http://ccl.northwestern.edu/netlogo/index.shtml</a>

Literatur:	<p>The following literature will be extended during the lectures:</p> <ul style="list-style-type: none"><li>• Weiss, Gerhard (ed.): Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence</li><li>• Mohri Mehryar et al: Foundations of Machine Learning</li><li>• Bonabeau, Eric et al: Swarm Intelligence – From Natural to Artificial Systems</li><li>• Brueckner Sven et al (ed.): Engineering Self-Organising Systems: Methodologies and Applications</li><li>• Shen, Weiming et al: Multi-Agent Systems for concurrent Intelligent Design and Manufacturing</li></ul>
------------	---

Modulbezeichnung:	<b><i>Organic Computing</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Bernhard Sick, Dr.-Ing. Sven Tomforde
Sprache:	Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS: 2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h: 60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	<ul style="list-style-type: none"> <li>- Grundlagen der Informatik</li> <li>- Einführung der Programmierung</li> </ul>
Angestrebte Lernergebnisse:	<p>Der/die Lernende kann:</p> <ul style="list-style-type: none"> <li>- Grundprinzipien der Selbstorganisation und Selbstadaption in technischen Systemen erklären,</li> <li>- Aspekte wie Emergenz, Robustheit und Selbstorganisation quantifizieren,</li> <li>- intelligente technische Systeme gemäß Organic Computing Ansätzen planen, entwerfen und entwickeln, und</li> <li>- die Verfahren zur Umsetzung der Adaptivität in Organic Computing Systemen vergleichen und bewerten.</li> </ul>
Inhalt:	<ul style="list-style-type: none"> <li>- Komplexität in technischen Systemen</li> <li>- Selbstorganisation</li> <li>- Quantifizierung von Systemeigenschaften (Emergenz, Selbstorganisation, Robustheit)</li> <li>- Entwurf von einzelnen Organic Computing Systemen</li> <li>- Entwurf kollaborativer Organic Computing Systeme</li> <li>- Modellierung von Organic Computing Systemen</li> <li>- Steuerung von Organic Computing Systemen</li> <li>- Anytime Learning</li> <li>- Anwendungsbeispiele</li> </ul>
Studien-/Prüfungsleistungen:	<p>Studienleistungen: Übungen</p> <p>Prüfungsleistungen: mündliche Prüfung (25 Minuten) oder schriftliche Prüfung (120 Minuten)</p>
Medienformen:	Medien: Folien, Tafel, Übungsblätter, wissenschaftliche Veröffentlichungen
Literatur:	<p>Die grundlegende Vorlesung basiert auf den folgenden Büchern, weiterführende Literatur wird im Rahmen der einzelnen Themen benannt:</p> <ul style="list-style-type: none"> <li>- C. Müller-Schloer, H. Schmeck, T. Ungerer (eds.): Organic Computing – A Paradigm Shift for Complex Systems</li> <li>- R. Würtz (ed.): Organic Computing</li> </ul>

	<ul style="list-style-type: none"><li>- P. Lalanda, J. McCann, A. Diaconescu: Autonomic Computing – Principles, Design and Implementation</li><li>- E. Alpaydin: Introduction to Machine Learning</li><li>- G. Di Marzo Serugendo, M.-P. Gleizes, A. Karageorgos (Eds.): Self-organising Software - From Natural to Artificial Adaptation</li></ul>
--	---

Modulbezeichnung:	<b><i>Pattern Recognition</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Sick und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse Stochastik, Analysis und lineare Algebra
Angestrebte Lernergebnisse:	Der/die Lernende kann: - verschiedene Aufgaben, Modelle und Algorithmen der Mustererkennung erklären, - neue Modellierungsansätze für Klassifikations- und Regressionsprobleme entwickeln, - neue Anwendungen eigenständig planen und realisieren, - existierende Verfahren und Anwendungen kritisch hinterfragen, vergleichen und bewerten.
Inhalt:	Die Vorlesung beschäftigt sich mit Grundlagen und Verfahren der Mustererkennung insbesondere aus einer probabilistischen Sichtweise. Folgende Themen werden besprochen: Grundlagen (u.a. Stochastik, Modellselektion, CurseofDimensionality, Entscheidungs- und Informatonstheorie), Verteilungen (u.a. Multinomial-, Dirichlet-, Gauss- und Student-Verteilung, Nichtparametrische Schätzung), Lineare Modelle für Regression, Lineare Modelle für Klassifikation, Mischmodelle und ExpectationMaximization, Approximative Inferenz, Kombination von Modellen, Statistische Lerntheorie (Support Vector Machines), Beispielanwendungen (Online-Clustering, Anomalieerkennung u.a.)
Studien-/Prüfungsleistungen:	Studienleistungen: regelmäßige Bearbeitung von Übungsaufgaben Prüfungsleistungen: mündliche Prüfung (20 Min.)
Medienformen:	Präsentation mit Beamer, Papierübungen
Literatur:	- Vorlesungsfolien, - verschiedene Kapitel des Buches Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer (2006), - zur Ergänzung auch Auszüge aus dem Buch Richard O. Duda, Peter E. Hart, David G. Stork: Pattern Classification, Wiley & Sons; 2. Auflage (2000), - weitere Literatur wird in der Vorlesung bekannt gegeben.

Modulbezeichnung:	<b><i>Seminar im Bereich Praktische Informatik</i></b>
Studiensemester:	Wintersemester 2016/2017 Sommersemester 2017
Modulverantwortliche/r	Dozenten des Bereichs Praktische Informatik
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	Seminar / 2 SWS
Arbeitsaufwand:	120 h:           ca. 30 h Präsenzzeit ca. 90 h Selbststudium
Kreditpunkte:	4 CP
Empfohlene Voraussetzungen:	Abhängig vom gewählten Seminar
Angestrebte Lernergebnisse:	Der/die Lernende kann: - sich ein Informatikthema selbständig aus der Literatur, inklusive englischsprachiger Originalarbeiten erschließen - im Umgang mit Fachtexten wissenschaftliche Arbeitsmethoden anwenden, z.B. das Hinterfragen von Aussagen, das Bilden eigener Urteile, das Überprüfen von Aussagen, das Hinzuziehen von Sekundärliteratur und das Zusammenfügen von Informationen aus unterschiedlichen Quellen - wissenschaftliche Inhalte für Präsentationen verständlich und strukturiert aufbereiten - in wissenschaftlichen Präsentationen geeignete mündliche und schriftliche Ausdrucksformen einsetzen.
Inhalt:	Abhängig vom konkreten Seminar
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistung: Vortrag (30 – 45 Min.) und Hausarbeit (max. 20 Seiten) oder Vortrag (max. 90 Min.)
Medienformen:	Abhängig vom konkreten Seminar
Literatur:	Abhängig vom konkreten Seminar

Modulbezeichnung:	<b><i>Software Engineering II</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Zündorf und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der/die Lernende kann: - einen interaktiven Diagrammeditor für eine vorgegebene grafische Modellierungssprache entwerfen und implementieren. Dies umfasst ein Metamodell und eine grafische Benutzeroberfläche, - Konsistenzanalysen für das Metamodell entwerfen und implementieren, die fehlerhafte Benutzereingaben aufdecken, - einen Interpreter und / oder Code Generator für die grafische Modellierungssprache entwerfen und implementieren.
Inhalt:	Am Beispiel eines einfachen graphischen Editors wie z.B. eines Statechart Editors wird eine Referenzarchitektur für interaktive graphische Tools vorgestellt. Danach werden die einzelnen Komponenten dieser Architektur und deren typischen Implementierungsvarianten und relevanten Designentscheidungen vorgestellt. Jeder Teilnehmer baut dann (in einer kleinen Gruppe) eine eigene Realisierung der betrachteten Komponente. Am Schluss der Vorlesung bzw. Projektarbeit hat dann jeder Teilnehmer ein lauffähiges interaktives Tool wie z.B. einen Statechart Editor inklusive z.B. einer Code-Generierung.
Studien-/Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Implementierung eines grafischen Modellierungswerkzeugs inklusive Metamodell, grafischer Benutzeroberfläche, Konsistenzanalysen, und Interpreter oder Code-Generator für eine vorgegebene grafische Modellierungssprache
Medienformen:	Tafel, Folien, Programmdemonstrationen, Vorlesungsbegleitende Webseite, Selbststudium
Literatur:	Wird in der Vorlesung bekannt gegeben

Modulbezeichnung:	<b><i>Temporal and Spatial Data Mining</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Sick und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        3 SWS Vorlesung 1 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Grundkenntnisse Stochastik, Analysis und lineare Algebra.
Angestrebte Lernergebnisse:	Der/die Lernende kann: - verschiedene Aufgaben, Modelle und Algorithmen des Spatio-Temporal Data Mining erklären, - neue Modellierungsansätze für Probleme wie Zeitreihenklassifikation, Anomalieerkennung, Motiverkennung u.a. entwickeln, - neue Anwendungen eigenständig planen und realisieren existierende Verfahren und Anwendungen kritisch hinterfragen, vergleichen und bewerten.
Inhalt:	Die Vorlesung beschäftigt sich mit Grundlagen der Mustererkennung in Zeitreihen (z.B. Sensorsignale) und räumlich verteilt erfassten Daten (z.B. in Sensornetzen). Es werden u.a. folgende Themen besprochen: - Grundlagen (z.B. Segmentierung von Zeitreihen, Korrelation von Daten, Merkmale zur Beschreibung temporaler/räumlicher Daten), - Abstandsmessung von Zeitreihen, Clustering/Klassifikation, Motiverkennung, Anomalieerkennung mit verschiedenen Techniken (z.B. NearestNeighbor, Neuronale Netze, Support Vector Machines), - verschiedenste Beispielanwendungen (Unterschriftenverifikation, kollaborative Gefahrenwarnung in Fahrzeugen, Aktivitätserkennung und Kontextererkennung mit Smartphones u.a.).
Studien-/Prüfungsleistungen:	Studienleistungen: Beteiligung an der Übung durch Kurzreferate zu ausgewählten Verfahren Prüfungsleistungen: mündliche Prüfung ( 20 Min.)
Medienformen:	Präsentation mit Beamer und Overhead, Papierübungen und Rechnerübungen
Literatur:	Folien zur Vorlesung, Auszüge aus folgenden Büchern: - T. Mitsa: Temporal Data Mining, Chapman & Hall / CRC (2010), - J. Gama: Knowledge Discovery from Data Streams, Chapman & Hall / CRC (2010), - S. Shekhar: Spatial and Spatiotemporal Data Mining, Chapman & Hall / CRC (2010),

	- weitere Literatur zu bestimmten Algorithmen wird in der Vorlesung bekannt gegeben.
--	--

Modulbezeichnung:	<b><i>Verteilte Systeme – Basisalgorithmen / Distributed Computing Algorithms</i></b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Geihs und Mitarbeiter
Sprache:	Englisch und Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	<p>The attendend is able to:</p> <ul style="list-style-type: none"> <li>• explain given distributed algorithms and analyse their properties,</li> <li>• evaluate the complexity of algorithms, develop extensions for given algorithms, implement distributed algorithms,</li> <li>• determine the applicability of given algorithms to new application scenarios.</li> </ul> <p>Der/die Lernende kann:</p> <ul style="list-style-type: none"> <li>• vorgegebene bzw. bekannte verteilte Algorithmen erklären und ihre Eigenschaften analysieren,</li> <li>• die Komplexität der Algorithmen qualitativ beurteilen, Erweiterungen für vorhandene Algorithmen entwickeln, verteilte Algorithmen implementieren,</li> <li>• die Anwendbarkeit vorgegebener verteilter Algorithmen in neuen Anwendungsszenarien bestimmen.</li> </ul>
Inhalt:	<p>The course deals with algorithmic and system independent foundations of distributed systems. Part of the topics are causality, logical clocks, distributed synchronisation, distributed deadlock recognition, peer-to-peer, fault tolerance, group communication, etc.</p> <p>Die Vorlesung behandelt die algorithmischen, systemunabhängigen Grundlagen verteilter Systeme. Zu den Themen gehören Kausalität, logische Uhren, verteilte Synchronisation, verteilte Deadlock-Erkennung, Peer-to-Peer, Fehlertoleranz, Gruppenkommunikation, etc.</p>
Studien-/Prüfungsleistungen:	<p>Studienleistungen: Exercises/ Übungen Prüfungsleistungen: oral exam / mündliche Prüfung (25 Minuten) or written exam / schriftliche Prüfung (120 Minuten)</p>
Medienformen:	<p>Media: Slides, Board, Course Associated Web Page (<a href="http://www.vs.uni-kassel.de">www.vs.uni-kassel.de</a>)</p> <p>Medienformen: Folien, Tafel, vorlesungsbegleitende Web Page (<a href="http://www.vs.uni-kassel.de">www.vs.uni-kassel.de</a>)</p>

Literatur:	<ul style="list-style-type: none"><li>• Coulouris, G., Dollimore, J. Und Kindberg, T.: Distributed Systems, 3. Aufl., Addison-Wesley 2000,</li><li>• Mattern, F.: Verteilte Basisalgorithmen, Springer-Verlag 1989,</li><li>• Tanenbaum, A. und van Stehen, M.: Distributed Systems, Prent. Hall 2002,</li></ul> <p>Further litarture will be announced during the lectures. / Weitere Literatur wird in der Vorlesung bekannt gegeben.</p>
------------	---

Modulbezeichnung:	<b>Web Engineering</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Zündorf und Mitarbeiter
Sprache:	Deutsch/Englisch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180 h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Datenbanken, Java
Angestrebte Lernergebnisse:	Die Studierenden können einfache Web-Anwendungen wie zum Beispiel einen Web Shop sowohl Serverseitig als auch Clientseitig konzipieren, realisieren, aufsetzen und betreiben. Die Studierenden können auch komplexe, interaktive Web-Anwendungen wie einen Diagrammeditor konzipieren, realisieren, aufsetzen und betreiben.
Inhalt:	Techniken für klassische Three Tier Web Applikationen mit Datenbank Layer, Business Logik und Web-Oberfläche. Entwicklung interaktiver Web Anwendungen mit Javascript und Model View Controller Mechanismen.
Studien-/Prüfungsleistungen:	Studienleistungen: Hausaufgaben Prüfungsleistungen: Projektaufgabe
Medienformen:	Folien, Web-Sites
Literatur:	Wird in der Vorlesung bekannt gegeben

Modulbezeichnung:	<b>Web Science</b>
Studiensemester:	Sommersemester 2017
Dozent(in):	Prof. Dr. Stumme und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Praktische Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	
Angestrebte Lernergebnisse:	Der/die Lernende kann: neben den inhaltlichen Konzepten der einzelnen Inhaltsbereiche der Vorlesung sowohl grundlegende als auch fortgeschrittene Techniken und Algorithmen wiedergeben, bewerten und anwenden.
Inhalt:	Die Vorlesung gibt eine Einführung in das neue Forschungsgebiet „Web Science“, das vor allem die Gebiete Social Semantic Web, Web 2.0, Knowledge Management, Social Network Analysis, Information Retrieval und Web Mining verbindet.
Prüfungsleistungen:	Studienleistungen: keine Prüfungsleistungen: Klausur oder mündliche Prüfung (120 bzw. 30 Minuten)
Medienformen:	
Literatur:	Wird in der Vorlesung bekannt gegeben.

## 5. Module im Bereich Vertiefung in Theoretischer, Technischer oder Praktischer Informatik

Für den Vertiefungsbereich darf frei aus allen Modulen der Theoretischen, Technischen und Praktischen Informatik gewählt werden. Zusätzlich ist das folgende Modul wählbar:

Modulbezeichnung:	<b><i>Modellgestützte Fabrikplanung</i></b>
Studiensemester:	Wintersemester 2016/2017
Dozent(in):	Prof. Dr. Wenzel und Mitarbeiter
Sprache:	Deutsch
Zuordnung zum Curriculum:	Vertiefung in Theoretischer, Technischer oder Praktischer Informatik
Lehrform/SWS:	4 SWS:        2 SWS Vorlesung 2 SWS Übung
Arbeitsaufwand:	180h:         60h Präsenzzeit 120h Selbststudium
Kreditpunkte:	6 CP
Empfohlene Voraussetzungen:	Statistikkenntnisse
Angestrebte Lernergebnisse:	<p>Der / die Lernende kann</p> <ul style="list-style-type: none"> <li>- system- und modelltheoretische Grundlagen zur Modellbildung und Simulation sowie die ereignisdiskrete Simulation als modellgestützte Analysemethode und ihre grundlegenden Zusammenhänge verständlich erklären,</li> <li>- Anwendbarkeit der Simulation für eine konkrete Aufgabenstellung bewerten,</li> <li>- ein Vorgehensmodell zur Simulation nachvollziehbar begründen,</li> <li>- selbständig Simulationsmodelle entwickeln, verifizieren und validieren, analysieren und statistisch abgesicherte Simulationsergebnisse erzeugen und interpretieren,</li> <li>- konkrete Fallbeispiele der Fabrikplanung untersuchen und basierend auf den Simulationsergebnissen Rückschlüsse auf das zu untersuchende System ziehen,</li> <li>- eigenständig die Erkenntnisse auf ähnlich gelagerte Aufgaben übertragen.</li> </ul>
Inhalt:	<p>Die Veranstaltung umfasst den Einsatz der ereignisdiskreten Simulation bei der Planung von Produktions- und Logistikanlagen sowie die konkrete Anwendung eines am Markt eingesetzten Simulationswerkzeuges zur Durchführung kleiner Simulationsstudien.</p> <ul style="list-style-type: none"> <li>- Folgende Themen werden im Einzelnen behandelt: system- und modelltheoretische Grundlagen</li> <li>- Bediensysteme,</li> <li>- analytische Berechnungsverfahren für ausgewählte Fragestellungen in der Fabrikplanung; Abgrenzung zu simulationsgestützten Verfahren,</li> </ul>

	<ul style="list-style-type: none"> <li>- Stochastik: Wahrscheinlichkeitsbegriff, Zufallszahlen, diskrete und stetige Zufallsgrößen, Wahrscheinlichkeitsverteilungen, Fragen der Anwendung,</li> <li>- Stimationsmethoden / Schedulingstrategien und Modellierungskonzepte ,</li> <li>- Vorgehensmodelle der Simulation: Konzeptuelles und formales Modell, Datenmanagement, Validierung und Verifikation, Experimentplanung, Ergebnisaufbereitung/-interpretation</li> <li>- Überblick über Simulationswerkzeuge in Produktion und Logistik,</li> <li>- Beispiele für Industriefanwendungen, Grundregeln und Checklisten.</li> </ul> <p>Die begleitenden Übungen dienen der praktischen Anwendung eines Simulationswerkzeugs. Dabei liegt der Schwerpunkt auf der Modellerstellung und der Analyse der Ergebnisse im Hinblick auf ein vorgegebenes Untersuchungsziel.</p>
Studien-/Prüfungsleistungen:	<p>Studienleistung: keine  Prüfungsleistungen: schriftliche Prüfung (90 Min.)</p>
Medienformen:	<p>Tafel, Folien, vorlesungsbegleitende Unterlagen, Arbeiten mit Simulationsprogrammen am Rechner, Selbststudium</p>
Literatur:	<p>Die folgende Literaturliste ist Grundlage der Veranstaltung; sie wird jedoch laufend aktualisiert und ergänzt:</p> <ul style="list-style-type: none"> <li>- Arnold, D.; Furmans, K.: Materialfluss in Logistiksystemen. Springer, Berlin, 2005,</li> <li>- Fahrmeir, et al: Statistik. 3. Auflage, Springer, Berlin, 2003,</li> <li>- Law, A.M.: Simulation Modeling and Analysis. 4. Auflage, McGraw-Hill, Boston, 2007,</li> <li>- Rabe, M., Spieckermann, S., Wenzel, S.: Verifikation und Validierung. VDI Springer, Berlin, 2008,</li> <li>- Robinson, S: Simulation, The Practice of Model Development and Use, John Wiley&amp; Sons, Chichester, 2004,</li> <li>- VDI 3633, Simulation von Logistik-, Materialfluss- und Produktionssystemen, Beuth, Düsseldorf, Blatt 1 ff.,</li> <li>- Wenzel, et al.: Qualitätskriterien für die Simulation in Produktion und Logistik. VDI Springer, Berlin, 2008.</li> </ul>

## 6. Spezielle Module

Modulbezeichnung:	<b><i>Projekt in Theoretischer, Technischer oder Praktischer Informatik</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Modulverantwortliche/r	Fohry
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Projekt in Theoretischer, Technischer oder Praktischer Informatik
Lehrform/SWS:	Projektarbeit / ca. 4 SWS
Arbeitsaufwand:	240 h:ca.: 60 h Präsenzzeit ca. 180 h Selbständige / Team-Arbeit
Kreditpunkte:	8 CP
Empfohlene Voraussetzungen:	Abhängig vom gewählten Projekt
Angestrebte Lernergebnisse:	Der/die Lernende kann: <ul style="list-style-type: none"> <li>- neue Anwendungen konzipieren und realisieren</li> <li>- dabei Informatik- und Projektmanagementmethoden beurteilen, auswählen und anwenden</li> <li>- anspruchsvolle Probleme analysieren und selbständig sowie in Zusammenarbeit mit anderen Studierenden lösen</li> <li>- im Team Fragen der Arbeitsorganisation, aufgetretene Konflikte oder die Einordnung der eigenen Arbeit in wirtschaftliche und gesellschaftliche Zusammenhänge reflektieren und konstruktiv bearbeiten</li> <li>- (je nach gewähltem Projekt) sich bei Bedarf Informatik- bzw. Anwendungskenntnisse aus der Literatur oder durch Experimente erschließen</li> <li>- (je nach gewähltem Projekt) andere Studierende anleiten</li> </ul>
Inhalt:	Abhängig vom konkreten Projekt
Studien-/Prüfungsleistungen:	Studienleistung: abhängig vom konkreten Projekt Prüfungsleistung: Projektarbeit
Medienformen:	Abhängig vom konkreten Projekt
Literatur:	Abhängig vom konkreten Projekt

Modulbezeichnung:	<b><i>Seminar in Theoretischer, Technischer oder Praktischer Informatik</i></b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Modulverantwortliche/r	Fohry
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum:	Seminar in Theoretischer, Technischer oder Praktischer Informatik
Lehrform/SWS:	Seminar / 2 SWS
Arbeitsaufwand:	120 h:           ca. 30 h Präsenzzeit ca. 90 h Selbststudium
Kreditpunkte:	4 CP
Empfohlene Voraussetzungen:	Abhängig vom gewählten Seminar
Angestrebte Lernergebnisse:	Der/die Lernende kann: - sich ein Informatikthema selbständig aus der Literatur, inklusive englischsprachiger Originalarbeiten erschließen - im Umgang mit Fachtexten wissenschaftliche Arbeitsmethoden anwenden, z.B. das Hinterfragen von Aussagen, das Bilden eigener Urteile, das Überprüfen von Aussagen, das Hinzuziehen von Sekundärliteratur und das Zusammenfügen von Informationen aus unterschiedlichen Quellen - wissenschaftliche Inhalte für Präsentationen verständlich und strukturiert aufbereiten - in wissenschaftlichen Präsentationen geeignete mündliche und schriftliche Ausdrucksformen einsetzen.
Inhalt:	Abhängig vom konkreten Seminar
Studien-/Prüfungsleistungen:	Studienleistung: keine Prüfungsleistung: Vortrag (30 – 45 Min.) und Hausarbeit (max. 20 Seiten) oder Vortrag (max. 90 Min.)
Medienformen:	Abhängig vom konkreten Seminar
Literatur:	Abhängig vom konkreten Seminar

Modulbezeichnung:	<b>Schlüsselkompetenzen</b>
Studiensemester:	Wintersemester 2016/2017 und Sommersemester 2017
Modulverantwortliche/r	Fohry
Sprache:	Deutsch oder Englisch
Zuordnung zum Curriculum	Schlüsselkompetenzen
Lehrform/SWS:	Abhängig von den gewählten Veranstaltungen
Arbeitsaufwand:	Abhängig von den gewählten Veranstaltungen
Kreditpunkte:	Abhängig von den gewählten Veranstaltungen
Empfohlene Voraussetzungen:	Abhängig von den gewählten Veranstaltungen
Angestrebte Lernergebnisse:	<p>Der/die Lernende kann zusätzliche oder vertiefte Kenntnisse oder Fertigkeiten in ein oder zwei der folgenden Bereiche vorweisen:</p> <ul style="list-style-type: none"> <li>- Projektmanagement</li> <li>- Führungsqualifikation</li> <li>- Arbeits- und Organisationspsychologie</li> <li>- Interkulturelle Kommunikation</li> <li>- Wirtschaft</li> <li>- Recht</li> </ul>
Inhalt:	Abhängig von den gewählten Veranstaltungen
Studien-/Prüfungsleistungen:	Abhängig von den gewählten Veranstaltungen
Medienformen:	Abhängig von den gewählten Veranstaltungen
Literatur:	Abhängig von den gewählten Veranstaltungen