Trier University
of Applied Sciences

H O C H
S C H U L E
T R I E R

MODULHANDBUCH

der Master-Studiengänge im Fachbereich Informatik
Prüfungsordnung 2019

Inhaltsverzeichnis

Externe Module

Neben den oben genannten Modulen werden weitere Module von anderen Fachbereichen angeboten, welche in den Master-Studiengängen des Fachbereichs Informatik als Pflicht- oder Wahlpflichtmodul zur Verfügung stehen können:

Fachbereich Wirtschaft

- Architektur/Implementierung integrierter Systeme
- Data Warehouse
- Informationsmanagement
- Internet: Technologien und Anwendungen
- Operations Research

Fachbereich Technik

- Elektrodiagnostik
- Neuroprothetik
- Projektmanagement
- Simulationsverfahren

Abschlussarbeit			
Inhalte	Bearbeitung einer qualifizierten Aufgabenstellung aus der Praxis unter Anleitung. Hierbei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken eingeübt sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt.		
Lernergebnisse	Fähigkeit zur selbständigen Bearbeitung einer größeren Aufgabenstellung, deren Schwierigkeitsgrad der späteren Berufspraxis eines Master of Science entspricht.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Abhängig von der Aufgabenstellung; wird vom Betreuer festgelegt		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	30	30 Stunden	870 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Dekan des Fachbereichs Informatik		
Änderungsdatum	23.07.2019		

Advanced Game Technology			
Inhalte	Die Vorlesung befasst sich mit fortgeschrittenen Methoden der Bildsynthese, sowohl für die interaktive Darstellung, als auch für Offline Rendering. Sie orientiert sich stark am aktuellen Forschungsstand in der Computergrafik und umfasst folgende Themenschwerpunkte: ▪ Physikalische Grundlagen der Lichtausbreitung ▪ Reflexionseigenschaften und Materialmodelle ▪ Bildbasierte Techniken ▪ Fotorealismus ▪ Prozedurale Modellierung ▪ Computeranimation ▪ Volumetrische Effekte und Participating Media		
Lernergebnisse	Die Teilnehmer der Lehrveranstaltung lernen den aktuellen Forschungsstand im Bereich Computergrafik kennen. Sie erlangen die Fähigkeit, aktuelle Entwicklungen zu verstehen und umzusetzen, sowie zukünftige technologische Trends zu analysieren und kritisch zu reflektieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Bachelor-Module „Einführung in die Computergrafik" und „C/C++-Programmierung"		
Studienleistung	☒ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Rezk-Salama		
Modulverantwortliche(r)	Prof. Dr. C. Rezk-Salama		
Änderungsdatum	11.02.2019		

Anforderungsmanagement			
Inhalte	In den meisten Unternehmen sind Anforderungen an Softwaresysteme oft unklar, widersprüchlich, unvollständig und nicht nachvollziehbar dokumentiert. Existierende Anforderungsspezifikationen (Lasten- und Pflichtenhefte) sind veraltet. Wichtige Anforderungen werden oft zu spät erkannt oder sogar übersehen. Darüber hinaus werden Anforderungen oft qualitativ unzureichend formuliert und lassen Spielraum für Interpretation. Die Folgen: unzufriedene Kunden, explodierende Kosten, weit überschrittene Projekttermine, unwartbare Systeme. Aufgabe des Anforderungsmanagements (engl.: Requirements Engineering oder RE) ist es, aus oft vagen und teilweise widersprüchlichen Ideen eine möglichst vollständige, korrekte widerspruchs- und redundanzfrei, nachverfolgbare und atomare Systemspezifikationen zu erzeugen, um diesen aufgeführten Problemen frühzeitig entgegenwirken zu können.		
Lernergebnisse	Die Studierenden sollen aufbauend auf der gleichnamigen Bachelor-Vorlesung die Kenntnisse bzgl. der Modellierung, Ermittlung und Dokumentation von Anforderungen vertiefen. Die Studierenden sollen ▪ Systeme hinsichtlich Ihrer wesentlichen Eigenschaften erfassen und abstrahieren können, ▪ Anforderungen (funktional, nicht-funktional) an Systeme sammeln und spezifizieren können, ▪ Techniken zur Stakeholderidentifikation und Interviews kennen und anwenden können, ▪ Reviewarten kennen und anwenden können, ▪ Anforderungen auf deren Qualität beurteilen können, ▪ Agile Vorgehensweisen im Requirementsengineering kennen und anwenden können, und ▪ aktuelle Entwicklungen wie Standardisierungen im Anforderungsmanagement (z.B. ReqIF) kennen. Im zweiten Teil der Vorlesung wird auf aktuelle Entwicklungen im Bereich des Anforderungsmanagement eingegangen. Hierbei wird der Zusammenhang zwischen Anforderungsmanagement und Variantenmanagement behandelt. Im Mittelpunkt steht zunächst das Verstehen der variantenbezogenen Entwicklungstechniken (Domänen- und Applikationsengineering). Im Anschluss sollen die Studierenden aktuelle Werkzeuge aus dem Bereich Variantenmanagement kennenlernen und diese im Requirementsengineering anwenden. Das Wissen, Verstehen und die Anwendung werden durch praxisorientierte Übungen, sowie den Einsatz aktueller Anforderungsmanagement- und Variantenmanagement-Werkzeuge vertieft.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Grundlagen des Anforderungsmanagements“		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. G. Rock		
Modulverantwortliche(r)	Prof. Dr. G. Rock		
Änderungsdatum	28.09.2020		

Architektur von Cloud-Anwendungen		
Inhalte	Wer bisher Cloud Computing mit einem Dienst wie Dropbox gleichgesetzt hat, bei dem es lediglich um das Speichern von Daten „in der Cloud" geht, wird in diesem Modul eines Besseren belehrt. In diesem Modul wird deutlich, dass Cloud Computing die Möglichkeit bietet, komplette Anwendungen, die traditionell in den Rechenzentren von Firmen (und Hochschulen) betrieben werden, „in die Cloud" zu verlagern. Aber auch neu gegründeten Firmen bietet Cloud Computing eine gute Basis zum Starten ihrer Geschäfte ohne größere Investitionen zum Aufbau einer Rechnerinfrastruktur. Viele Firmen wie z.B. Netflix nutzen diese Möglichkeit heute schon sehr intensiv. Im Mittelpunkt dieses Moduls stehen exemplarisch die Cloud-Dienste von Amazon, die als Amazon Web Services (AWS) bezeichnet werden. Es wird vor allem darum gehen, die zahlreichen Dienste, die AWS bietet, kennenzulernen und darauf aufbauend komplette Anwendungen zu entwerfen und zu realisieren. Solche Anwendungen können u.a. aus Web-Servern, Anwendungs-Servern, Datenbanken (relational und NoSQL), Caching-Diensten, DNS-Servern usw. bestehen. Die AWS-Dienste ermöglichen es darüber hinaus, mit verhältnismäßig geringem Aufwand die Anwendungen hochverfügbar und fehlertolerant auszulegen. So gibt es die Möglichkeit, sowohl Lastbalancierungskomponenten in die eigene Anwendung zu integrieren als auch eine automatische Skalierung, so dass bei hoher Last zusätzliche Server-Instanzen gestartet und diese bei zurückgehender Last wieder heruntergefahren werden. Weitere Schwerpunkte sind die Bereiche Sicherheit und Kosteneffizienz. U.a. werden folgende Themen behandelt, wobei bei allen Themen die AWS-Dienste als Beispiele herangezogen werden: ▪ geschichtliche Entwicklung des Cloud Computing ▪ Kategorien des Cloud Computing: IaaS, PaaS, SaaS ▪ Vorteile des Cloud Computing ▪ Cloud-Dienste: virtuelle Maschinen, Blockspeicher, verteilte Dateisysteme, relationale und nicht-relationale Datenbanken, Objektspeicher, … ▪ Sicherheitskonzepte des Cloud Computing ▪ Hochverfügbarkeit, Lastverteilung und Fehlertoleranz im Cloud Computing ▪ Architektur größerer Cloud-Anwendungen ▪ Entwurfsmuster und Best Practices	
Lernergebnisse	Die Studierenden besitzen nach dem Absolvieren dieses Moduls folgende Fähigkeiten: ▪ Die Studierenden können die grundlegenden Konzepte des Cloud Computing sowie deren Vor- und Nachteile erläutern. ▪ Die Studierenden können einen Überblick über die unterschiedlichen Cloud-Dienste anhand der AWS-Dienste von Amazon geben. Sie können die Besonderheiten dieser Dienste, aber auch ihre Begrenzungen darstellen. ▪ Die Studierenden sind in der Lage, komplette, problemspezifische Anwendungsarchitekturen zu entwerfen und diese mit Hilfe der AWS-Dienste aufzubauen. Dabei werden Entwurfsmuster und Best Practices verwendet, so dass die Aspekte Sicherheit, Hochverfügbarkeit, Lastverteilung und Fehlertoleranz besondere Berücksichtigung finden. ▪ Die Studierenden können ihre eigenen Lösungen einem Publikum gegenüber präsentieren sowie die vorgestellten Lösungen ihrer Mitstudierenden kritisch bewerten.	
Literatur	▪ Originalunterlagen von Amazon AWS (werden Studierenden zur Verfügung gestellt).	
Lehrform	☒ Vorlesung	
	☒ Übung	
	☐ Seminar/Seminaristischer Unterricht	
	☐ Labor	
	☐ Projekt	
Empfohlene Voraussetzungen	Grundkenntnisse der Informatik, die in der Regel in einem Bachelor-Studium der Informatik vermittelt werden.	
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung	
	☐ Regelmäßige Teilnahme an den Übungen	
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten	
	☐ Bestehen von Leistungsstandkontrollen	
Prüfungsform	☐ Schriftliche Prüfung	
	☒ Mündliche Prüfung	
	☐ Prüfung am PC	
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar	
Verwendbarkeit	Informatik	☐ PF ☒ WPF

Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. R. Oechsle		
Modulverantwortliche(r)	Prof. Dr. R. Oechsle		
Änderungsdatum	23.07.2020		

Berechenbarkeit und Komplexität			
Inhalte	▪ Äquivalente Modelle der Computer-Berechenbarkeit: WHILE, Turingmaschinen ▪ Algorithmenbegriff, These von Church ▪ Berechenbare Funktionen und ihre grundlegenden Eigenschaften, z.B. s-m-n-Theorem, universelle Algorithmen, Gödelisierung ▪ Entscheidbarkeit, Semi-Entscheidbarkeit Aufzählbarkeit, äquivalente Charakterisierungen, die Klassen REC und RE ▪ Unentscheidbare Mengen, Halteproblem, Many-one-Reduzierbarkeit, RE-Vollständigkeit, Beispiel nicht-aufzählbarer Mengen, Reduktionsmethode, Satz von Rice ▪ Laufzeitanalyse, Polynomial- und Exponentialzeit, Katalog klassischer Entscheidungsprobleme ▪ Die Klassen P und NP, NP-Vollständigkeit, originärer Vollständigkeitsbeweis ▪ Reduktionen durch Komponentendesign, Vollständigkeitsbeweise für Auswahl-, Graphen- und Reihenfolgeprobleme ▪ Strenge NP-Vollständigkeit, Pseudo-Polynomialzeit, Large Number Problems, Dynamische Programmierung, pseudo-polynomielle Reduktionen ▪ NP-Optimierungsprobleme, die Klassen PO und NPO, Turing-Reduzierbarkeit, Begriff der (strengen) NP-Härte, PO-NPO-Frage ▪ Approximationsalgorithmen, Performanzrate, Beispiele mit konstanter Performanzrate, Approximierbarkeit, Approximationsschema		
Lernergebnisse	Die Studierenden können ▪ Berechenbarkeitsmodelle hinsichtlich ihrer prinzipiellen Leistungsfähigkeit einordnen, ▪ die Bedeutung eines präzisen Algorithmenbegriffs erläutern, ▪ Berechnungsprobleme hinsichtlich ihrer prinzipiellen algorithmischen Beherrschbarkeit analysieren und bewerten, ▪ die Bedeutung der P-NP-Theorie präsentieren und Berechnungsprobleme in dieser Hinsicht klassifizieren, ▪ die Methode der Reduktion auf neue Berechnungsprobleme anwenden und ▪ Optimierungsprobleme kategorisieren und deren praktische Lösbarkeit beurteilen.		
Literatur	▪ H. Rogers Jr.: Theory of Recursive Functions and Effective Computability. MIT–Press, 1967. 3rd printing 1992. ▪ Michael R. Garey and David S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., 1979. ▪ K. W. Wagner: Theoretische Informatik – Eine kompakte Einführung. Springer, 2nd edition, 2003.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	11.11.2019		

Data Science			

Inhalte	▪ Grundlagen und Anwendungsfälle für Data Science
	▪ Daten
	  ▪ typische Datenquellen im Unternehmen
	  ▪ Arten von Daten: Stammdaten, Bewegungsdaten
	  ▪ strukturierte, unstrukturierte und semistrukturierte Daten
	  ▪ Netzwerkdaten/Graphen
	  ▪ Werkzeuge zur Datenanalyse und -vorverarbeitung
	  ▪ Datenqualität
	▪ Datenverarbeitung
	  ▪ Architektur von Datenverarbeitungsstrecken
	  ▪ Umgang mit großen Datenmengen
	  ▪ Datenintegration
	  ▪ Data Ingestion
	  ▪ Präsentationsschicht
	▪ Business Intelligence
	  ▪ Grundlagen Data Warehousing
	  ▪ Reporting
	  ▪ Visualisierung
	▪ Empfehlungssysteme
	▪ Social Network Analysis		

Lernergebnisse	Die Studierenden sind nach der Teilnahme in der Lage ▪ vorhandene Datenbestände in Unternehmen zu bewerten, zu verarbeiten und zu nutzen, ▪ die Qualität und Struktur von Daten einzuschätzen, ▪ Daten aus verschiedenen Quellen aufzubereiten und zu integrieren, ▪ Daten und daraus abgeleitetes Wissen für Fachanwender zu präsentieren, ▪ gängige Werkzeuge wie Kafka, Camel, PDI, Pandas, Jupyter einzusetzen und ▪ Wissen und Empfehlungen aus sozialen Netzwerken zu generieren. Die Vorlesung ist im Zusammenspiel mit der Veranstaltung „Maschinelles Lernen" konzipiert, in der konkrete Lernverfahren vermittelt werden.		
Literatur	▪ Cady, F.: The Data Science Handbook. Wiley, 2017. ▪ O'Neil, C., Schutt, R.: Doing Data Science. O'Reilly, 2014. ▪ Kotu, V., Deshpande, B.: Data Science: Concepts and Practice. 2. Auflage, Morgan Kaufmann, 2019. ▪ Barabasi, A.-L., Posfai, M.: Network Science. Cambridge University Press, 2016. ▪ Aggarwal, C. C.: Recommender Systems: The Textbook. Springer, 2016. ▪ Kimball, R., Ross, M.: The Data Warehouse Toolkit. 3. Auflage, Wiley, 2013.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Datenbanken" und „Big-Data-Technologien"		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Schmitz		
Modulverantwortliche(r)	Prof. Dr. C. Schmitz		
Änderungsdatum	23.10.2019		

Fortgeschrittene Methoden der Computergrafik			
Inhalte	▪ Mathematische Grundlagen: homogene Koordinaten, Transformationen, Projektionen, Splines ▪ Vektor-Raster Konvertierung: Linien, Kurven, Kreise, Bresenham, Füllalgorithmen, Replikation, Antialiasing, Ausgabetechniken ▪ Geometrische Datenstrukturen: Kurven; Flächen, Volumina. ▪ Farbmodelle: chromatisches und achromatisches Licht, Gammakorrektur, CIE-Farbraum, RGB-, CMYK-, HSV-, HLS-Modell ▪ Rendering: Clipping, verdeckte Kanten, Ray-Tracing, Radiosity, etc.		
Lernergebnisse	Die Studierenden werden in den Grundlagen der grafischen Datenverarbeitung unterrichtet. In praktischen Übungen werden einfache Aufgaben der grafischen Datenverarbeitung realisiert und damit vertieft.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☒ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. F. N. Rudolph		
Modulverantwortliche(r)	Prof. Dr. F. N. Rudolph		
Änderungsdatum	27.08.2013		

Ganzzahlige Lineare Optimierung			
Inhalte	▪ Ganzzahlige Lineare Programme (MIP, IP, BIP) ▪ Beispielprobleme und deren Modellierung, Modellierungstechniken, Komplexitätsbetrachtung ▪ Formulierungen und optimale LP-Relaxierungen, vollständig unimodulare Matrizen ▪ LP-basiertes Branch-and-Bound-Verfahren ▪ Spaltengenerierung, Branch-and-Price ▪ Schnittebenenverfahren, Branch-and-Cut ▪ Praktischer Einsatz von LP-Solvern ▪ Laborprojekt		
Lernergebnisse	Die Studierenden können ▪ kombinatorische Optimierungsprobleme als MIPs modellieren, ▪ die zugrundeliegende Theorie der ganzzahligen Linearen Programme erläutern, ▪ die algorithmischen Verfahren darstellen und gegenüberstellen, ▪ die behandelten Lösungsmethoden für Beispielprobleme auswählen und anwenden sowie ▪ LP-Solver praktisch einsetzen und mit deren Hilfe schwierige Optimierungsprobleme lösen.		
Literatur	▪ Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. January 1997. ▪ Jiri Matousek and Bernd Gärtner: Understanding and Using Linear Programming. Universitext. Springer Science & Business Media, July 2007.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Kenntnisse der Linearen Optimierung (LP-Modellierung, Simplex-Verfahren, Dualität)		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	05.12.2019		

Geschäftsprozessmanagement			

Inhalte	▪ Definitionen und Begriffsklärungen im Kontext Geschäftsprozessmanagement (GPM)
	▪ Strategisches und operatives GPM
	▪ Phasen des sog. GPM-Zyklus
	▪ GPM-Modellierungsmethoden:
	  ▪ Petri-Netze als theoretische Modellierungsgrundlage von Geschäftsprozessen
	  ▪ Business Process Model and Notation (BPMN)
	  ▪ Business Process Execution Language (BPEL)
	▪ Standardisierungsansätze im GPM-Umfeld (WfMC, OMG, OASIS)
	▪ Implementierungsaspekte von GPM-Systemen, Service-orientierte Architekturen (SOA), Web Services
	▪ Process Monitoring und Process Mining		

Lernergebnisse	Die Studierenden sollen ▪ den Sinn und Einsatzpotentiale von GPM-Systemen als Mittel zur erfolgreichen Umsetzung eines ganzheitlichen Prozessmanagements verstehen, ▪ Geschäftsprozesse mit Standard-Notationen beschreiben können, ▪ wesentliche Prozessablaufmuster (Workflow Patterns) beschreiben und modellieren können, ▪ Funktionalitäten existierender GPM-Systeme kennen und kritisch beurteilen können, ▪ den GPM-Zyklus mit Hilfe eines GPM-Werkzeugs in den Übungen exemplarisch durchlaufen können.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. A. Lux		
Modulverantwortliche(r)	Prof. Dr. A. Lux		
Änderungsdatum	12.11.2018		

High Performance Computing			

Inhalte	▪ General paradigms for parallel programming
	▪ Using the Threading Building Blocks Library for CPU parallelism
	  ▪ Concepts of the task stealing scheduler
	  ▪ Efficient memory management for parallel systems
	  ▪ General parallelism concepts in TBB
	▪ Using CUDA for GPGPU computing
	  ▪ Concepts of GPGPU computing
	  ▪ Writing simple CUDA programs
	  ▪ Synchronization in CUDA
	  ▪ Streaming and overlapping in CUDA		

Lernergebnisse	Students should be able to decide which HPC concept to use for a specific task at hand for compute intensive operations. They should be able to implement those in concrete programs.		
Literatur	▪ Michael McCool, Arch D. Robinson, James Reinders: Structured Parallel Programming. Morgan Kaufmann, 2012. ▪ James Reinders: Intel Threading Building Blocks. O'Reilly, 2007. ▪ Jason Sanders, Edward Kandrot: CUDA by Example. Addison-Wesley Professional, 2010.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Competences according to the learning outcome of the bachelor modules „C/C++-Programmierung" (C/C++ Programming) and „Technische Informatik" (Computer Engineering, especially Computer Architecture).		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	28.10.2019		

Implementierung von ERP-Systemen			
Inhalte	▪ Architektur betrieblicher Informationssysteme am Beispiel eines verbreiteten Systems ▪ Erlernen einer anwendungsorientierten Programmiersprache ▪ Programmierung von Standardanwendungsfällen und Zugriffen auf ein SAP-Systemen		
Lernergebnisse	Die Studierenden lernen den konkreten Aufbau, die Konfiguration und die Realisierung eines komplexen betrieblichen Anwendungssystems (z. B. SAP) kennen. Sie verstehen die grundlegenden Zusammenhänge und können die Einsatzmöglichkeiten im betrieblichen Umfeld beurteilen. Die Studierenden können Standardanwendungsfälle innerhalb des Anwendungssystems realisieren.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Objektorientierte Programmierung - Einführung", „Datenbanken" und „Produktionswirtschaft"		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. F. N. Rudolph		
Modulverantwortliche(r)	Prof. Dr. F. N. Rudolph		
Änderungsdatum	28.09.2020		

Informationssicherheit			

Inhalte	▪ Einführung: grundlegende Begriffe und Zusammenhänge
	▪ Physische Sicherheit
	▪ Zugriffskontrolle
	  ▪ Authentisierungsverfahren und –techniken
	  ▪ Zugriffskontrollmodelle
	  ▪ Rechtevergabe
	▪ Organisatorische, konzeptionelle und prozessorientierte Sicherheit
	  ▪ Sicherheitskonzepte gemäß IT-Grundschutz des BSI
	  ▪ Sicherheitsmanagement und ISO 27000
	  ▪ Business Continuity Planning
	▪ Standards zur Informationssicherheit
	▪ Rechtliche Aspekte der Informationssicherheit
	▪ Computer und Internet-Kriminalität
	▪ Malware
	▪ Ggf. andere Themen wie Security Engineering, Patch Management		

Lernergebnisse	Die Studierenden sollen ▪ einen gesamtheitlichen Einblick in die Informationssicherheit erlernen und verstehen, dass IT-Sicherheit neben Technik insbesondere auch physische, organisatorische, konzeptionelle und rechtliche Aspekte beinhaltet. ▪ strukturierte Vorgehensweisen zur Umsetzung von Informationssicherheit anhand von Standards wie z.B. ISO 27x oder BSI 100 kennen und anwenden können. ▪ erlernen Risikoanalysen durchzuführen. ▪ die Historie von Computermalware und ausgewählte Malware im Detail kennen und Algorithmen zur Malwareerkennung kennen und in Beispielaufgaben anwenden können. ▪ fortgeschrittene Verfahren zur Authentisierung und Autorisierung kennen wie z.B. die biometrische Authentisierung über den Fingerabdruck oder RBAC und in Beispielaufgaben anwenden und berechnen können. ▪ die Grundprinzipien und Standards des Patch Managements kennen und in Beispielaufgaben anwenden können.		
Literatur	▪ Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons ▪ John Aycock, Computer Viruses and Malware, Springer ▪ Bundesamt für Sicherheit in der Informationstechnik, Standardreihe 200-x ▪ Claudia Eckert, IT-Sicherheit: Konzepte, Verfahren, Protokolle, Oldenbourg ▪ Shon Harris, CISSP Certification All-in-one Exam Guide: Complete coverage of all Certified Information Systems Security Professional domains, McGraw-Hill ▪ ISO 27001 und ISO 27002 ▪ Norbert Pohlmann: Cyber-Sicherheit, Springer, 2019Als eBook der Bibliothek vohanden ▪ Sowa: Management der Informationssicherheit, Springer		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor (Ausgewählter Vorlesungsstoff und die Übungen werden vertieft durch praktische Übungen.)		
	☐ Projekt		
Empfohlene Voraussetzungen	Besuch einer einführenden Veranstaltung zur IT-Sicherheit im Bachelor-Studium		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	75 Stunden	105 Stunden

Lehrende(r)	Prof. Dr. K. Knorr
Modulverantwortliche(r)	Prof. Dr. K. Knorr
Änderungsdatum	24.10.2019

Interactive Physical Simulation			

Inhalte	▪ Collision detection
	  ▪ Coarse granular collision detection (sweep and prune, BSP trees, Octrees)
	  ▪ Fine granular collision detection (primitive collision, hierarchical OBBS, Minkowski sums, k-dops)
	  ▪ Tunnel Problem and Solutions like conservative advancement and retroactive detection
	▪ physical properties of moving rigid bodies (force, torque, acceleration, momentum, impulse, velocity...), Newton mechanic
	▪ numerical Integration of ODEs, L- and A- stability, implicit semi implicit and explicit solvers.
	▪ Collision reaction
	  ▪ Collision reaction between two rigid bodies with one point of contact
	  ▪ Multiple rigid bodies and multiple points of contact
	  ▪ Impulse base Simulation methods
	▪ Contraint based Simulation methods
	  ▪ Early methods like penalty and verlet integration
	  ▪ Lagrange Dynamics for contrained motion
	  ▪ linear complimentary problems
	  ▪ Simulation of breaking and tearing for games with lagrange dynamics
	  ▪ Coupling of Simulation and Animation for articulated characters
	▪ Particle Simulation
	  ▪ Statefull and stateless particle simulation
	  ▪ Crowd / Panic Simulation with particles		

Lernergebnisse	Students should be able to understand the techniques used in physical simulators for games, apply the proper technique for the specific situation at hand and implement portions of the system themselves.		
Literatur	▪ H. R. Schwarz: Numerische Mathematik. Springer Vieweg, 2011. ▪ David H. Eberly: Game Physics. Taylor & Francis, 2. Auflage, 2010. ▪ Gino van den Bergen: Collision Detection in Interactive 3D Environments. CRC Press, 2003.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Competences according to the learning outcome of the bachelor modules „Lineare Algebra“ (Linear Algebra) and „Angewandte Mathematik“ (Applied Mathematics).		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☒ Mündliche Prüfung		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. C. Lürig		
Modulverantwortliche(r)	Prof. Dr. C. Lürig		
Änderungsdatum	28.10.2019		

Kooperative Systeme			
Inhalte	▪ Soziale und technische Aspekte der Kooperation ▪ Klassifikation von Groupware-Lösungen ▪ Gestaltung von Groupware-Lösungen ▪ Awareness, Kommunikations- und Koordinationsunterstützung ▪ Cognitive Systems Engineering ▪ Mensch-Maschine-Kooperation ▪ Geteilte Kontrolle und Entscheidungsunterstützung ▪ Symbolische und subsymbolische Ansätze künstlicher Intelligenz ▪ Intelligent User Interfaces ▪ Mensch-Maschine-Integration		
Lernergebnisse	Die Studierenden können ▪ Grundlagen, Prinzipien und Anwendungsmöglichkeiten computergestützter kooperativer Arbeit erläutern, ▪ die wichtigsten Theorien und Modelle der Mensch-Maschine-Kooperation beschreiben, und ▪ kooperative Systeme analysieren, konzipieren, realisieren und evaluieren.		
Literatur	▪ T. Gross & M. Koch: Computer-Supported Cooperative Work. Oldenbourg, 2007. ▪ D. Coleman: Groupware. Prentice-Hall, 1997. ▪ G. Schwabe et al. (Hrsg.): CSCW-Kompendium. Springer, 2001. ▪ M. Beaudouin-Lafon (Hrsg.): Computer-Supported Cooperative Work. Wiley, 1998. ▪ C. Reuter: Sicherheitskritische Mensch-Computer-Interaktion. Interaktive Technologien und Soziale Medien im Krisen- und Sicherheitsmanagement. 2018		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. T. Mentler		
Modulverantwortliche(r)	Prof. Dr. T. Mentler		
Änderungsdatum	03.08.2020		

Lineare Optimierung			
Inhalte	▪ Lineare Programme ▪ Beispielprobleme und deren Modellierung, Standardform, Slackform ▪ Simplexverfahren in Grundform, Nachweis der Korrektheit, Eindeutigkeit optimaler Lösungen, Initialisierung, Hauptsatz der Linearen Programmierung, ökonomische Interpretation der optimalen Slackform ▪ LP-Dualität, schwache und starke Dualität, Satz vom komplementären Schlupf, Interpretation der Dualisierung in ökonomischen Modellen ▪ Revidiertes Simplexverfahren, temporäre Schattenpreise, Duales Simplexverfahren ▪ Geometrie Linearer Programme, konvexe Polyeder ▪ Einsatz von LP-Solvern		
Lernergebnisse	Die Studierenden können ▪ Berechnungsprobleme als lineare Programme modellieren, insbesondere Strukturvariablen identifizieren und Bedingungen mathematisch formulieren, ▪ die behandelten Verfahren reproduzieren und deren Korrektheit und Laufzeit erklären, ▪ Lineare Programme mit den behandelten Verfahren lösen und die erzielten Lösungen interpretieren, ▪ das Konzept der LP-Dualität wiedergeben und die zugrundeliegende Theorie begründen sowie ▪ den Einsatz von LP-Solvern am Beispiel praktisch umsetzen.		
Literatur	▪ Vasek Chvatal: Linear programming. A Series of Books in the Mathematical Sciences, 1983. ▪ Dimitris Bertsimas and John N. Tsitsiklis: Introduction to Linear Optimization. January 1997.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Grundkenntnisse der linearen Algebra		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	11.11.2019		

Maschinelles Lernen			
Inhalte	Die Methoden des maschinellen Lernens stellen die Grundlage für die aktuelle Forschung und Entwicklungen im Bereich intelligenter Systeme dar und sind integraler Bestandteile einer großen Anzahl gegenwärtiger, industrieller Anwendungen. Hierbei werden Modelle anhand vorliegender Daten trainiert, um spezifische Aufgabenstellung zu lösen. ▪ Einführung und grundlegende Konzepte ▪ Datenvorverarbeitung / Merkmalsextraktion (Zeit- und Frequenzbereich) / Dimensionsreduktion ▪ Hauptachsentransformation (PCA) / Whitening ▪ Common Spatial Pattern (CSP) ▪ Sammon-Transformation ▪ Multidimensional Skalierung (MDS), T-SNE ▪ Entscheidungsbäume ▪ Bayes'sche Inferenz, Der optimale Klassifikator ▪ Gauß'sche Mischverteilungen, Maximum Likelihood und Expectation Maximization (EM) ▪ k-means, k-Nearest Neighour, Kerndichteschätzer ▪ Linear und nicht linear separierbare Probleme ▪ Quadratische Fehlerminimierung und Gradientenabstiegsverfahren ▪ Fischer lineare Diskriminanzanalyse (LDA) ▪ Linearer Klassifikator und generalisierte lineare Diskriminanzfunktionen ▪ Lineare und multiple lineare Regression ▪ Binär logistische Regressionsanalyse ▪ Support Vector Machines ▪ Dynamic Time Warping ▪ Hidden-Markov-Modell ▪ Empirische Evaluierung von Lernverfahren ▪ Neuronale Netze: ▪ Ein und mehrschichtige Perzeptronen ▪ Convolutional Neural Networks (CNN) ▪ Rekurrente Neuronale Netze, LSTM ▪ Autoencoder ▪ Zielfunktionen für unterschiedliche Anwendungsfälle ▪ Backpropagation ▪ Stochastisches Gradientenverfahren ▪ Methoden zur Initialisierung der Parameter ▪ Dropout, Batch-Normalisation, Regularisierung der Parameternorm ▪ Steuern von Hyperparametern ▪ Transfer-Learning ▪ Überblick zu weiteren Netzarchitekturen (ResNet, WaveNet, U-Net, GAN, Boltzmann-Maschinen, ...)		
Lernergebnisse	Die Teilnehmer der Lehrveranstaltung lernen den aktuellen Forschungsstand im Bereich Maschinelles Lernen kennen. Sie erlangen die Fähigkeit, aktuelle Entwicklungen zu verstehen/realisieren, zu analysieren und kritisch zu reflektieren.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium

	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Lohscheller, Prof. Dr. H.-P. Beise		
Modulverantwortliche(r)	Prof. Dr. J. Lohscheller		
Änderungsdatum	09.12.2019		

Mensch-Computer-Interaktion			
Inhalte	▪ Grundlegende Begriffe der Mensch-Computer-Interaktion: Usability, User Experience, Accessibility ▪ Menschliche Informationsverarbeitung und Handlungsprozesse ▪ Ein-/Ausgabegeräte und Interaktionstechnologien ▪ Menschzentrierte Entwicklungsprozesse: ISO 9241-210, Contextual Design, Scenario-Based Design ▪ Usability und Software Engineering ▪ Methoden zur Analyse von Nutzungskontexten: Hierarchische Aufgabenanalyse, Personas, Prozessanalyse ▪ Low- und High-Fidelity-Prototyping ▪ Formative und summative Evaluationen ▪ Modelle für Mensch-Computer-Systeme ▪ Interaktion und Kooperation ▪ Normen und rechtliche Grundlagen		
Lernergebnisse	Die Studierenden können ▪ Prinzipien und Methoden menschzentrierter Entwicklung interaktiver Systeme erläutern. ▪ Kenntnisse über die menschliche Informationsverarbeitung in die Entwicklung interaktiver Systeme einbringen. ▪ Modelle und Kriterien zur Analyse und Bewertung interaktiver Systeme anwenden.		
Literatur	▪ Benyon, D. (2014). Designing interactive systems: A comprehensive guide to HCI, UX and interaction design (Third edition). Harlow, England: Pearson. ▪ Dahm, M. (2006). Grundlagen der Mensch-Computer-Interaktion. Informatik: Software-Ergonomie. München: Pearson Studium. ▪ Herczeg, M. (2018). Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme: De Gruyter. ▪ Reuter, C. (Hrsg.) (2018). Sicherheitskritische Mensch-Computer-Interaktion. Wiesbaden: Springer Fachmedien Wiesbaden.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. T. Mentler		
Modulverantwortliche(r)	Prof. Dr. T. Mentler		
Änderungsdatum	31.01.2020		

Programm- und Systemanalyse			
Inhalte	▪ Einsatzgebiete für statische Programmanalyse und darauf aufbauende Systemanalysen ▪ Abgrenzung zu nicht semantikbasierten Ansätzen und dynamischer Programmanalyse ▪ Grundlegender Aufbau von Compilern ▪ Unentscheidbarkeit von Analyseaufgaben (Satz von Rice), konservative vs. optimistische Approximation ▪ Standardbeispiele Compileroptimierung (u.a. Available Expressions, Live Variables) ▪ Datenflussanalyse: Mathematische Grundlagen, Einführung in Programmanalysegenerator (PAG), Entwicklung eigener Analysen in einer domänspezifischen, funktionalen Programmiersprache ▪ Beispiele industriell eingesetzter Analysewerkzeuge (u.a. WCET-Analyse, Stackanalyzer, Astrée) ▪ Abstrakte Interpretation ▪ Schedulability Analyse als Beispiel für Systemanalysen ▪ Beispiele aus aktuellen Forschungsaktivitäten		
Lernergebnisse	Die Studierenden sollen die Prinzipien moderner Programm- und Systemanalsewerkzeuge kennenlernen. Sie sollen in der Lage sein zu beurteilen, welche Ansätze und Mechanismen für konkrete Aufgaben besser oder schlechter geeignet sind. Für verfügbare Analysewerkzeuge sollen sie die Voraussetzungen für deren Anwendbarkeit und deren Grenzen einschätzen können. Darüber hinaus sollen sie die mathematischen Grundlagen der Datenflussanalyse erlernen, eigene Datenflussanalysen entwickeln können und die Grundprinzipien der Abstrakten Interpretation kennen lernen.		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Theoretische Informatik“, fundierte Kenntnisse über die Prinzipien von Programmiersprachen		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Schneider		
Modulverantwortliche(r)	Prof. Dr. J. Schneider		
Änderungsdatum	12.08.2013		

Projektstudium			
Inhalte	Bearbeitung einer qualifizierten Aufgabenstellung aus der Praxis unter Anleitung. Hierbei werden systematische Vorgehensweisen und sinnvolle Arbeitstechniken eingeübt sowie die Verbindung zu Anwendungsgebieten der Informatik hergestellt.		
Lernergebnisse	Fähigkeit zur selbständigen Bearbeitung einer größeren Aufgabenstellung, deren Schwierigkeitsgrad der späteren Berufspraxis eines Master of Science entspricht.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☒ Projekt		
Empfohlene Voraussetzungen	Abhängig von der Aufgabenstellung; wird vom Betreuer festgelegt		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	15	25 Stunden	425 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Dekan des Fachbereichs Informatik		
Änderungsdatum	23.07.2019		

Seminar			
Inhalte	Selbständiges Erarbeiten eines vorgegebenen begrenzten Themenbereiches anhand von Fachliteratur und anderen Quellen sowie dessen schriftliche und mündliche Darstellung. Es werden wechselnde aktuelle Themen aus der Informatik angeboten, die im Schwierigkeitsgrad für den Master-Studiengang Informatik angemessen sind.		
Lernergebnisse	Fähigkeit zur selbständigen Erarbeitung eines Themenbereiches und dessen angemessene und verständliche Darstellung.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Abhängig vom Thema des Seminars; wird vom Lehrenden festgelegt		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☒ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	3	20 Stunden	70 Stunden
Lehrende(r)	Dozenten des Fachbereichs Informatik		
Modulverantwortliche(r)	Dekan des Fachbereichs Informatik		
Änderungsdatum	23.07.2019		

Software-Architekturen			
Inhalte	▪ Architekturmuster (z.B. Microservices) ▪ Entwurf von Architekturen ▪ Dokumentation von Architekturen ▪ Bewertung von Architekturen ▪ Model-Driven-Architecture™ (MDA) ▪ Enterprise Integration Patterns		
Lernergebnisse	Die Studierenden sollen ▪ konkrete Architekturen analysieren und verstehen können, ▪ passende Architekturen und Architekturmuster auf Problemstellungen anwenden können, ▪ Architekturen beschreiben und entwerfen können, ▪ Analyse- und Entwurfsmuster anwenden können, ▪ spezifische Verfahren zum modellbasierten Architekturentwurf anwenden können und ▪ klassische n-tier und moderne Architekturen kennen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse der Module „Objektorientierte Programmierung - Einführung", „Software-Entwurf" und „Grundlagen des Anforderungsmanagements"		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. Christoph Schmitz		
Modulverantwortliche(r)	Prof. Dr. Christoph Schmitz		
Änderungsdatum	28.09.2020		

Software-Qualitätsmanagement			

Inhalte	▪ Einführung und Überblick
	▪ Qualitätssicherung
	▪ Manuelle Prüfmethoden
	▪ Verbesserung der Prozessqualität
	▪ Produktqualität - Komponenten
	  ▪ Testende Verfahren
	  ▪ Verifizierende Verfahren
	  ▪ Analysierende Verfahren
	▪ Produktqualität - Systeme
	  ▪ Integrationstest
	  ▪ System- und Abnahmetest
	▪ Konfigurations- und Änderungsmanagement		

Lernergebnisse	Die Studierenden sollen ▪ Qualitätsbegriffe definieren und einordnen können ▪ die Prinzipien der Software-Qualitätssicherung erklären und begründen können ▪ (Code-)Inspektionen durchführen können ▪ kontrollflussorientierte und datenflussorientierte Testverfahren einsetzen können ▪ die Konzepte der Verifikation und des symbolischen Testens verwenden und gegen testende Verfahren abgrenzen können ▪ für einfache Beispiele Integrations- und Abnahmetests durchführen können ▪ Testwerkzeuge einsetzen können ▪ Werkzeuge und Verfahren des Konfigurationsmanagements einsetzen können		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Keine		
Studienleistung	☒ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☒ Schriftliche Prüfung		
	☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☒ PF ☐ WPF
Angebot	☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. G. Rock		
Modulverantwortliche(r)	Prof. Dr. G. Rock		
Änderungsdatum	13.06.2019		

Ubiquitous Computing			
Inhalte	Die folgenden Themen werden als Aspekte von Systemen im Bereich „Ubiquitous Computing" betrachtet: ▪ Beispielsysteme aus Forschung und Anwendung ▪ Erhebung, Verwendung und Verteilung von Kontextinformationen ▪ „Location" als besonderer Kontext ▪ Anpassung von Anwendungen an Benutzer und Situation ▪ Alternative Ein- und Ausgabe-, sowie Interaktionsmöglichkeiten ▪ Aufbau und Durchführung von Benutzerstudien ▪ Umgang und Unterstützung von Privacy als Kernproblematik von Ubicomp Systemen ▪ Drahtlose Datenkommunikation und Ad-hoc Protokolle ▪ Spracheingabe und Sprachausgabe ▪ Zukünftige Entwicklungen Weiterhin werden wissenschaftliche Papiere aus den relevanten wissenschaftlichen Konferenzen und Journalen diskutiert, um die Themen der Vorlesung an aktuellen Forschungsergebnissen zu spiegeln.		
Lernergebnisse	Studierende haben Kenntnis vom Bereich „Ubiquitous Computing", sowie den zugehörigen Anwendungen, Konzepten, Verfahrensweisen und aktuellen Forschungsthemen. Sie analysieren Situationen aus dem Blickwinkel des Forschungsgebietes und sind in der Lage aus den Beobachtungen und unter Verwendung der gängigen Methoden und Vorgehensweisen des Gebiets ein UbiComp-System zu konzipieren, zu erstellen und zu evaluieren. Sie sind in der Lage ihre Ergebnisse als wissenschaftliches Papier zu formulieren, im Stil, wie er im Forschungsgebiet üblich ist.		
Literatur	▪ John Krumm : Ubiquitous Computing Fundamentals. Taylor & Francis Ltd, 2009. ▪ Wissenschaftliche Papiere der Proceedings der Konferenzreihe UbiComp, bzw. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) Journal werden im Rahmen der Übung vorgestellt und diskutiert.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Web-Technologien". Die Veranstaltung ist teilnahmebeschränkt. Bei Überbuchung werden die Plätze durch einen Einstufungstest vergeben.		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☐ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☒ Mündliche Prüfung		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. G. Schneider		
Modulverantwortliche(r)	Prof. Dr. G. Schneider		
Änderungsdatum	23.10.2019		

Verlässliche Echtzeitsysteme			
Inhalte	▪ Dependability ▪ Terminologie (z.B. nach Laprie) ▪ Echtzeitsysteme ▪ Systemeigenschaft Sicherheit ▪ Einschlägige Normen (z.B. IEC 61508, ISO 26262, DO 178 B) ▪ Fehlervermeidung vs. Fehlertoleranz ▪ Formale Methoden ▪ Echtzeitbetriebssysteme ▪ Parallelisierung von Echtzeitsystemen ▪ Anwendungsbeispiele, z.B. ESP, Motorsteuerung, Fahrerassistenzsysteme		
Lernergebnisse	Die Studierenden sollen die Prinzipien verlässlicher Echtzeitsysteme und die Besonderheiten bei deren Entwicklung kennen lernen. Sie sollen in der Lage sein in interdisziplinären Teams bei der Entwicklung von sicherheitsrelevanten Echtzeitsystemen mitzuwirken.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Technische Informatik"		
Studienleistung	☐ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☐ Mündliche Prüfung		
	☐ Prüfung am PC		
	☒ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. J. Schneider		
Modulverantwortliche(r)	Prof. Dr. J. Schneider		
Änderungsdatum	28.09.2020		

Verteilte Systeme			
Inhalte	In diesem Modul werden wichtige Konzepte verteilter Systeme anhand „klassischer" Originalartikel besprochen. Im Vorlesungsteil werden die wesentlichen Inhalte einer oder mehrerer verwandter Arbeiten vom Dozenten vorgestellt. In den Übungen beantworten die Studierenden Fragen zu den in der Vorlesung präsentierten Sachverhalten, erarbeiten anhand der Literatur Themen, die in der Vorlesung nicht oder nur oberflächlich behandelt wurden, und erstellen dafür eine Präsentation. Ferner implementieren sie einige der behandelten Algorithmen. Die behandelten Themen sind: ▪ Einleitung ▪ Verteilte Graph-Algorithmen ▪ Peer-to-Peer-Systeme (CAN, Chord, Pastry/Tapestry, P-Grid) ▪ Terminierungserkennung (Termination Detection) und Abfallsammlung (Garbage Collection) ▪ Verklemmungserkennung ▪ Logische und reale Zeit ▪ Nachrichtenreihenfolge ▪ Konsensfindung und Koordination ▪ Globale Zustände ▪ Globale Prädikate ▪ Transaktionen, insbesondere Nebenläufigkeitskontrolle (2-Phasen-Sperren mit Verklemmungsbehandlung, Zeitstempel-Verfahren, optimistische Verfahren) ▪ Datenreplikation		
Lernergebnisse	▪ Die Studierenden können grundlegende Probleme verteilter Systeme erläutern und dafür grundlegende Lösungsansätze darstellen (s. Inhalte). ▪ Die Studierenden können sich mit wissenschaftlichen Artikeln (in der Regel in Englisch) aus Zeitschriften und Konferenzen auseinandersetzen, wesentliche Informationen daraus extrahieren und diese für eine Präsentation in gut verständlicher Form aufbereiten. Dabei diskutieren sie die Inhalte kritisch und gehen auf ihre Verständnisprobleme bzw. aus ihrer Sicht entdeckter Ungereimtheiten in den Artikeln ein. ▪ Die Studierenden können verteilte Algorithmen, die in wissenschaftlichen Artikeln in Pseudocode beschrieben werden, in einer Programmiersprache und -umgebung ihrer Wahl implementieren.		
Literatur	▪ Im Wesentlichen: Mehrere Originalartikel (werden in der Vorlesung bekannt gegeben). Dazu als Hintergrundinformation: ▪ A. Tanenbaum, M. Van Steen: Distributed Systems – Principles and Paradigms, 2nd Edition, 2007. ▪ G. Coulouris, J. Dollimore, T. Kindberg, G. Blair: Distributed Systems – Concepts and Design, 6th Edition, 2013. ▪ A. Kshemkalyani, M. Singhal: Distributed Computing – Principles, Algorithms, and Systems, 2011.		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Informatik, die in der Regel in einem Bachelor-Studium der Informatik vermittelt werden.		
Studienleistung	☒ Regelmäßige Teilnahme an der Vorlesung		
	☒ Regelmäßige Teilnahme an den Übungen		
	☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten		
	☐ Bestehen von Leistungsstandkontrollen		
Prüfungsform	☐ Schriftliche Prüfung		
	☒ Mündliche Prüfung		
	☐ Prüfung am PC		
	☐ Hausarbeit/Projekt mit Kolloquium/Seminar		
Verwendbarkeit	Informatik		☐ PF ☒ WPF
Angebot	☐ Sommersemester ☐ Wintersemester ☒ Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte	Kontaktzeit	Selbststudium
	6	60 Stunden	120 Stunden
Lehrende(r)	Prof. Dr. R. Oechsle		
Modulverantwortliche(r)	Prof. Dr. R. Oechsle		
Änderungsdatum	23.07.2020		