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		

Algorithmik			

Inhalte	▪ Optimierungs- und Entscheidungsprobleme, NP-harte Optimierungsprobleme
	▪ Verschiedene Ansätze zur Lösung praktisch relevanter, NP-harter Optimierungsprobleme:
	  ▪ Deterministische Verfahren, Pseudo-Polynomialzeit, parametrisierte Komplexität, lineare Programmierung
	  ▪ Approximationen mit Gütegarantie: Design-Techniken für Approximationsalgorithmen, Klassen der Approximierbarkeit
	  ▪ Randomisierte Algorithmen
	  ▪ Heuristiken
	▪ Kombination von Verfahren, vom Problem zum Algorithmus, Projektbeispiele		

Lernergebnisse	Die Studierenden sollen ▪ einige typische kombinatorische Optimierungsprobleme kennen lernen und ihre praktische Lösbarkeit beurteilen können ▪ verschiedene Konzepte zur Lösung NP-harter Probleme kennen und anwenden lernen ▪ Algorithmen entwerfen und hinsichtlich ihrer Laufzeit und Lösungsgüte beurteilen können ▪ Experimentelle Analysen von Algorithmen durchführen können		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Berechenbarkeit und Komplexität“		
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. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	12.11.2018		

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 vertieft und weiter ausgebaut werden. Die Vorlesung behandelt hierbei Themen wie: ▪ Systemanalyse ▪ Anforderungsmanagement ▪ Validierung von Anforderungen ▪ Ziel- und Szenariobasiertes RE ▪ Agile Vorgehensweisen im RE ▪ Aktuelle Entwicklungen wie Standardisierungen im Anforderungsmanagement (z.B. ReqIF) 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 stehen die Ermittlung, Spezifikation und Verifikation von Varianten mit Hilfe von Feature Modellen. Wie in der Bachelor-Vorlesung werden die Erkenntnisse durch praxisorientierte Übungen, sowie den Einsatz kommerzieller Anforderungsmanagement- und Variantenmanagement-Werkzeuge vertieft.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Anforderungsmanagement“		
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		

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 erfahren Sie, 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 sollen nach dem Absolvieren dieses Moduls folgende Fähigkeiten besitzen: • Sie sollen die grundlegenden Konzepte des Cloud Computing kennen sowie die Vorteile des Cloud Computing benennen können. • Sie sollen einen Überblick über die unterschiedlichen Cloud-Dienste anhand der AWS-Dienste von Amazon geben können. Sie sollen die Besonderheiten dieser Dienste, aber auch ihre Begrenzungen darstellen können. • Sie sollen in der Lage sein, komplette, problemspezifische Anwendungsarchitekturen zu entwerfen und diese mit Hilfe der AWS-Dienste aufzubauen. Dabei sollen Entwurfsmuster und Best Practices verwendet werden, so dass die Aspekte Sicherheit, Hochverfügbarkeit, Lastverteilung und Fehlertoleranz besondere Berücksichtigung finden.		
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	13.06.2019		

<table>
<tr><td colspan="5">Berechenbarkeit und Komplexität</td></tr>
<tr><td>Inhalte</td><td colspan="4">▪ Äquivalente Modelle der Computer-Berechenbarkeit: Programmiersprache RIES, Turingmaschinen
▪ Algorithmenbegriff, These von Church
▪ Berechenbare Funktionen und ihre Eigenschaften
▪ Entscheidbarkeit, Aufzählbarkeit, unentscheidbare Mengen, Halteproblem
▪ Definition von Komplexitätsklassen (Laufzeit, Speicherplatz)
▪ P-NP-Problem, Theorie der NP-Vollständigkeit
▪ Wichtige Komplexitätsklassen und ihre Beziehungen</td></tr>
<tr><td>Lernergebnisse</td><td colspan="4">Die Studierenden sollen
▪ Berechenbarkeitsmodelle hinsichtlich ihrer prinzipiellen Leistungsfähigkeit einordnen können
▪ die Bedeutung eines präzisen Algorithmenbegriffs kennenlernen
▪ Berechnungsprobleme hinsichtlich ihrer prinzipiellen und praktischen Lösbarkeit einordnen können
▪ Komplexitätsmaße für Algorithmen bestimmen können
▪ die Komplexität von Berechnungsproblemen bewerten können</td></tr>
<tr><td rowspan="5">Lehrform</td><td colspan="4">☒ Vorlesung</td></tr>
<tr><td colspan="4">☒ Übung</td></tr>
<tr><td colspan="4">☐ Seminar/Seminaristischer Unterricht</td></tr>
<tr><td colspan="4">☐ Labor</td></tr>
<tr><td colspan="4">☐ Projekt</td></tr>
<tr><td>Empfohlene Voraussetzungen</td><td colspan="4">Keine</td></tr>
<tr><td rowspan="4">Studienleistung</td><td colspan="4">☐ Regelmäßige Teilnahme an der Vorlesung</td></tr>
<tr><td colspan="4">☒ Regelmäßige Teilnahme an den Übungen</td></tr>
<tr><td colspan="4">☒ Regelmäßige Bearbeitung von Haus-/Laborarbeiten</td></tr>
<tr><td colspan="4">☐ Bestehen von Leistungsstandkontrollen</td></tr>
<tr><td rowspan="4">Prüfungsform</td><td colspan="4">☒ Schriftliche Prüfung</td></tr>
<tr><td colspan="4">☒ Mündliche Prüfung (nur bei geringer Teilnehmerzahl)</td></tr>
<tr><td colspan="4">☐ Prüfung am PC</td></tr>
<tr><td colspan="4">☐ Hausarbeit/Projekt mit Kolloquium/Seminar</td></tr>
<tr><td>Verwendbarkeit</td><td colspan="3">Informatik</td><td>☒ PF ☐ WPF</td></tr>
<tr><td>Angebot</td><td colspan="4">☐ Sommersemester ☒ Wintersemester ☐ Unregelmäßig</td></tr>
<tr><td rowspan="2">Arbeitsaufwand</td><td>ECTS-Punkte</td><td>Kontaktzeit</td><td colspan="2">Selbststudium</td></tr>
<tr><td>6</td><td>60 Stunden</td><td colspan="2">120 Stunden</td></tr>
<tr><td>Lehrende(r)</td><td colspan="4">Prof. Dr. H. Schmitz</td></tr>
<tr><td>Modulverantwortliche(r)</td><td colspan="4">Prof. Dr. H. Schmitz</td></tr>
<tr><td>Änderungsdatum</td><td colspan="4">12.11.2018</td></tr>
</table>

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 sollen in der Lage sein, ▪ 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 ▪ 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.		
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	23.08.2019		

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, Komplexitätsbetrachtung ▪ Formulierungen und LP-Relaxierungen ▪ Branch-and-Bound-Verfahren ▪ Spaltengenerierung ▪ Branch-and-Cut, Branch-and-Price ▪ Praktischer Einsatz von LP-Solvern ▪ Laborprojekt		
Lernergebnisse	Die Studierenden sollen ▪ kombinatorische Optimierungsprobleme als MIPs modellieren können, ▪ Grundkenntnisse über die zugrundeliegende Theorie erwerben, um passende ▪ Lösungsmethoden auswählen und anwenden zu können, ▪ den praktischen Einsatz von LP-Solvern kennen lernen.		
Lehrform	☒ Vorlesung		
	☒ Übung		
	☐ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Grundkenntnisse der Python-Programmierung, Kenntnisse der Lineare 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		
	☐ 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	27.03.2019		

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.		
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	27.03.2014		

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", „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	12.11.2018		

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.		
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	12.11.2018		

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.		
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	12.11.2018		

Komponentenbasierte und generative Software-Entwicklung			
Inhalte	▪ Frameworks zur Entwicklung web-basierter Anwendungen (Java Server Faces, Struts) ▪ Objekt-relationales Mapping (Hibernate, Java Persistence Architecture) ▪ Frameworks für Anwendungs-Server (Enterprise Java Beans) ▪ Frameworks zur Realisierung von Dependency Injection und aspekt-orientierter Programmierung (Spring) ▪ Modellgetriebene Software-Entwicklung (MDSD / MDA)		
Lernergebnisse	Das Hauptziel dieser Vorlesung ist die Vermittlung wesentlicher Bestandteile moderner Software-Architekturen. Dies bedeutet u.a., dass die Entwicklung großer Software-Systeme heute nicht mehr „auf der grünen Wiese" beginnt, sondern dass es zum einen für einige Anwendungsbereiche bereits mächtige Basissysteme (Frameworks) gibt, für die dann anwendungsspezifische Komponenten (Plugins) dazu entwickelt werden (komponentenbasierte Software-Entwicklung), und dass zum anderen Programmcode in vielen Fällen aus Konfigurationsdateien oder kommentarartigen Anmerkungen in Programmen (z.B. Annotationen in Java) automatisch erzeugt wird (generative Software-Entwicklung). In dieser Lehrveranstaltung werden wichtige Konzepte wie Dependency Injection, aspekt-orientierte Programmierung, modellgetriebene Software-Entwicklung und objekt-relationales Mapping von den Studierenden erarbeitet und für die Lösung praxisnaher Problemstellungen eingesetzt.		
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 (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. R. Oechsle		
Modulverantwortliche(r)	Prof. Dr. R. Oechsle		
Änderungsdatum	12.11.2018		

Lineare Optimierung			
Inhalte	▪ Lineare Programme ▪ Beispielprobleme und deren Modellierung ▪ Simplexverfahren in Grundform, revidiertes Simplexverfahren ▪ LP-Dualität, duales Simplexverfahren ▪ Geometrie von LPs, konvexe Polyeder ▪ Einsatz von LP-Solvern		
Lernergebnisse	Die Studierenden sollen ▪ Berechnungsprobleme als LPs modellieren können, ▪ Grundkenntnisse über die zugrundeliegende Theorie erwerben, um passende Lösungsmethoden auswählen und anwenden zu können, ▪ LPs von Hand mit den behandelten Verfahren lösen können und die Lösungen interpretieren können ▪ den praktischen Einsatz von LP-Solvern am Beispiel kennenlernen.		
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	09.10.2019		

Maschinelles Lernen
Die Modulbeschreibung wird nachgereicht.

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		

<table>
<tr><td colspan="5">Software-Architekturen</td></tr>
<tr><td>Inhalte</td><td colspan="4">▪ Architekturmuster (z.B. Microservices)
▪ Entwurf von Architekturen
▪ Dokumentation von Architekturen
▪ Bewertung von Architekturen
▪ Model-Driven-Architecture™ (MDA)
▪ Enterprise Integration Patterns</td></tr>
<tr><td>Lernergebnisse</td><td colspan="4">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.</td></tr>
<tr><td rowspan="5">Lehrform</td><td colspan="4">☒ Vorlesung</td></tr>
<tr><td colspan="4">☒ Übung</td></tr>
<tr><td colspan="4">☐ Seminar/Seminaristischer Unterricht</td></tr>
<tr><td colspan="4">☐ Labor</td></tr>
<tr><td colspan="4">☐ Projekt</td></tr>
<tr><td>Empfohlene Voraussetzungen</td><td colspan="4">Kompetenzen gemäß der Lernergebnisse der Module „Objektorientierte Programmierung", „Software-Entwurf" und „Grundlagen des Anforderungsmanagement"</td></tr>
<tr><td rowspan="4">Studienleistung</td><td colspan="4">☐ Regelmäßige Teilnahme an der Vorlesung</td></tr>
<tr><td colspan="4">☒ Regelmäßige Teilnahme an den Übungen</td></tr>
<tr><td colspan="4">☐ Regelmäßige Bearbeitung von Haus-/Laborarbeiten</td></tr>
<tr><td colspan="4">☐ Bestehen von Leistungsstandkontrollen</td></tr>
<tr><td rowspan="4">Prüfungsform</td><td colspan="4">☒ Schriftliche Prüfung</td></tr>
<tr><td colspan="4">☐ Mündliche Prüfung</td></tr>
<tr><td colspan="4">☐ Prüfung am PC</td></tr>
<tr><td colspan="4">☐ Hausarbeit/Projekt mit Kolloquium/Seminar</td></tr>
<tr><td>Verwendbarkeit</td><td colspan="3">Informatik</td><td>☒ PF ☐ WPF</td></tr>
<tr><td>Angebot</td><td colspan="4">☒ Sommersemester ☐ Wintersemester ☐ Unregelmäßig</td></tr>
<tr><td rowspan="2">Arbeitsaufwand</td><td>ECTS-Punkte</td><td>Kontaktzeit</td><td colspan="2">Selbststudium</td></tr>
<tr><td>5</td><td>60 Stunden</td><td colspan="2">90 Stunden</td></tr>
<tr><td>Lehrende(r)</td><td colspan="4">Prof. Dr. Christoph Schmitz</td></tr>
<tr><td>Modulverantwortliche(r)</td><td colspan="4">Prof. Dr. Christoph Schmitz</td></tr>
<tr><td>Änderungsdatum</td><td colspan="4">07.10.2019</td></tr>
</table>

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	▪ Drahtlose Datenkommunikation ▪ Ermittlung von Kontextinformation (Sensoren, Benutzertracking) ▪ Spracheingabe und Sprachausgabe ▪ Alternative Eingabe- und Interaktionsmöglichkeiten ▪ Protokolle ▪ Anpassung von Anwendungen an Benutzer und Situation ▪ Benutzerstudien ▪ Privacy Weiterhin werden wissenschaftliche Papiere aus den relevanten wissenschaftlichen Konferenzen diskutiert.		
Lernergebnisse	Die Vermittlung von Kenntnissen über Anforderungen und Realisierung von ubiquitären Applikationen. Die Studierenden werden Randbedingungen für mobile Anwendungen kennen lernen (beschränktes Display, beschränkte Bandbreite, beschränkte Aufmerksamkeit des Benutzers) und Konzepte mobile Applikationen unter Berücksichtigung dieser Randbedingungen umzusetzen. Weiterhin werden sie Konzepte zur Realisierung von ubiquitären, heterogenen, kontextverarbeitenden Anwendungen kennen lernen. Typische Aufgaben für Informatiker sind hier z.B. das Design der Architektur und die Entwicklung, z.B. von „Location-based Services" für mobile Endgeräte oder Mobile User Interfaces für verteilte Anwendungen.		
Lehrform	☐ Vorlesung		
	☐ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☐ Labor		
	☐ Projekt		
Empfohlene Voraussetzungen	Kompetenzen gemäß der Lernergebnisse des Moduls „Webtechnologien". 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	12.04.2018		

Verifikation nebenläufiger Software-Systeme			
Inhalte	▪ Notwendigkeit der Verifikation von Software, Beispiele, Abgrenzung zum Software-Test ▪ Formale Verifikation durch Model Checking ▪ Korrektheitsnachweise für endliche, nebenläufige Zustandsmodelle ▪ Formale Spezifikation von Korrektheitsanforderungen mit Hilfe temporaler Logik ▪ Funktionsweise und praktischer Einsatz von Werkzeugen ▪ Laborprojekt: Modellierung, Spezifikation und Verifikation eines Protokolls oder Systems		
Lernergebnisse	Die Studierenden sollen ▪ Möglichkeiten und Grenzen der Verifikation einschätzen lernen ▪ moderne Methoden der automatischen Verifikation verstehen und anwenden können ▪ sich Funktionsweise und Einsatz von Werkzeugen selbstständig erarbeiten können ▪ nebenläufige Systeme modellieren und Anforderungen formal spezifizieren können		
Lehrform	☐ Vorlesung		
	☒ Übung		
	☒ Seminar/Seminaristischer Unterricht		
	☒ Labor		
	☐ Projekt		
e 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. H. Schmitz		
Modulverantwortliche(r)	Prof. Dr. H. Schmitz		
Änderungsdatum	30.01.2017		

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 der Module „Digitaltechnik" und „Rechnerarchitektur"		
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		

Verteilte Systeme			
Inhalte	▪ Zeit und globale Zustände in verteilten Systemen, Schnappschuss-Algorithmen, Algorithmen zur Abfallsammlung (Garbage Collection) und Terminierungserkennung (Termination Detection) ▪ Transaktionen, insbesondere Nebenläufigkeitskontrolle (2-Phasen-Sperren mit Verklemmungsbehandlung, Zeitstempel-Verfahren, optimistische Verfahren) ▪ Peer-to-Peer-Systeme (CAN, Chord, Pastry / Tapestry, P-Grid)		
Lernergebnisse	In dieser Vorlesung erarbeiten sich die Studierenden wichtige Konzepte verteilter Systeme anhand wissenschaftlicher Original-Artikel (in der Regel in Englisch) aus Zeitschriften und Konferenzen. Der Inhalt und die verwendeten Artikel variieren von Jahr zu Jahr (die aktuellen Inhalte s.o.). Neben der Durchdringung der Konzepte ist gleichberechtigtes Lernziel, dass die Studierenden in die Lage versetzt werden, wissenschaftliche Artikel lesen, verstehen und kritisch diskutieren zu können (dabei auch Ungereimtheiten oder Fehler entdecken). Zur Vertiefung des Stoffes müssen unterschiedliche Aufgaben von den Studierenden gelöst werden.		
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	19.01.2017		