

Modulhandbuch

Master-Verbund-Studiengang

Angewandte Informatik

FPO März 2020

Stand: Sommersemester 2024

Studienverlaufsplan Start Wintersemester

	1. Semester Wintersemester	2. Semester Sommersemester	3. Semester Wintersemester	4. Semester Sommersemester	5. Semester Wintersemester	6. Semester Sommersemester
Pflichtmodule aller Studienrichtungen	Kryptographie <i>6 ECTS</i>	Funktionale und logische Programmierung <i>6 ECTS</i>	Master- Konferenzseminar <i>6 ECTS</i>	Compilerbau <i>6 ECTS</i>	Netzökonomie <i>6 ECTS</i>	Masterarbeit <i>20 ECTS</i>
	IT-Vertragsrecht <i>6 ECTS</i>	NoSQL-Datenbanken <i>6 ECTS</i>	Wahlpflichtfach 1 <i>6 ECTS</i>	Wahlpflichtfach 2 <i>6 ECTS</i>	Projekt <i>18 ECTS</i>	Kolloquium <i>4 ECTS</i>
Pflichtmodule Studien- richtung „Anwendungs- entwicklung“	Modellgetriebene Software-Entwicklung <i>6 ECTS</i>	Vertiefung Software- Engineering <i>6 ECTS</i>	Moderne Web Frameworks <i>6 ECTS</i>	Usability Engineering <i>6 ECTS</i>		
Pflichtmodule Studienrichtung „Systemintegration“	Server-Betriebs- systeme <i>6 ECTS</i>	Technische Dokumentation <i>6 ECTS</i>	Netzwerksicherheit <i>6 ECTS</i>	Cloud Computing <i>6 ECTS</i>		
Pflichtmodule Studienrichtung „Künstliche Intelligenz“	Data Mining <i>6 ECTS</i>	Machine Learning <i>6 ECTS</i>	Big Data Processing <i>6 ECTS</i>	Deep Learning Anwendungen <i>6 ECTS</i>		

Studienverlaufsplan Start Sommersemester

	1. Semester Sommersemester	2. Semester Wintersemester	3. Semester Sommersemester	4. Semester Wintersemester	5. Semester Sommersemester	6. Semester Sommersemester
Pflichtmodule aller Studienrichtungen	Funktionale und logische Programmierung <i>6 ECTS</i>	Kryptographie <i>6 ECTS</i>	Compilerbau <i>6 ECTS</i>	Master- Konferenzseminar <i>6 ECTS</i>	Projekt <i>18 ECTS</i>	Netzökonomie <i>6 ECTS</i>
	NoSQL-Datenbanken <i>6 ECTS</i>	IT-Vertragsrecht <i>6 ECTS</i>	Wahlpflichtfach 2 <i>6 ECTS</i>	Wahlpflichtfach 1 <i>6 ECTS</i>	Masterarbeit <i>20 ECTS</i>	Kolloquium <i>4 ECTS</i>
Pflichtmodule Studien- richtung „Anwendungs- entwicklung“	Vertiefung Software- Engineering <i>6 ECTS</i>	Modellgetriebene Software-Entwicklung <i>6 ECTS</i>	Usability Engineering <i>6 ECTS</i>	Moderne Web Frameworks <i>6 ECTS</i>		
Pflichtmodule Studienrichtung „Systemintegration“	Technische Dokumentation <i>6 ECTS</i>	Server-Betriebs- systeme <i>6 ECTS</i>	Cloud Computing <i>6 ECTS</i>	Netzwerksicherheit <i>6 ECTS</i>		
Pflichtmodule Studienrichtung „Künstliche Intelligenz“	Machine Learning <i>6 ECTS</i>	Data Mining <i>6 ECTS</i>	Deep Learning Anwendun- gen <i>6 ECTS</i>	Big Data Processing <i>6 ECTS</i>		

Inhalt

Studienverlaufsplan Start Wintersemester	2
Studienverlaufsplan Start Sommersemester	3
Container	5
Pflichtmodule aller Studienrichtungen	6
Kryptographie.....	6
IT-Vertragsrecht.....	8
Funktionale und logische Programmierung.....	10
NoSQL-Datenbanken.....	12
Master-Konferenzseminar.....	14
Compilerbau.....	16
Netzökonomie	18
Projekt.....	20
Pflichtmodule Anwendungsentwicklung	22
Modellgetriebene Software-Entwicklung	23
Vertiefung Software-Engineering	25
Moderne Web-Frameworks.....	27
Usability Engineering	29
Pflichtmodule Systemintegration	31
Server-Betriebssysteme.....	32
Technische Dokumentation.....	34
Netzwerksicherheit.....	36
Cloud Computing	38
Pflichtmodule Künstliche Intelligenz	40
Data Mining.....	41
Machine Learning	43
Big Data Processing	45
Deep Learning Anwendungen.....	47
Masterarbeit und Kolloquium	49
Masterarbeit.....	50
Kolloquium	51
Wahlpflichtfächer / Containermodule	52
Data Science Anwendungsfälle (Container Datenanalyse).....	53
Effiziente Algorithmen (Container Algorithmen)	55
IT-Management (Container Betriebswirtschaft).....	57
Microsoft-Server-Betriebssysteme (Container Verteilte Systeme und Betriebssysteme).....	59
Praktische Anwendungen von Algorithmen (Container Algorithmen)	61
Web-Mining (Container Datenanalyse)	62
Wissensbasierte Systeme.....	64

Container

Folgende Container sind diesem Studiengang zugeordnet:

- Algorithmen
- Anwendungsgebiete der Informatik
- Betriebswirtschaft
- Datenanalyse
- Datenbanksysteme
- Gesellschaftliche Themen
- Grafische Datenverarbeitung
- Internet der Dinge
- IT-Recht und Datenschutz
- IT-Sicherheit
- Künstliche Intelligenz
- Programmiersysteme
- Softwareengineering
- Theoretische Informatik
- Verteilte Systeme und Betriebssysteme
- Vorgehensmodelle / IT-Projektmanagement

Die Module, die den einzelnen Containern zugeordnet sind, finden Sie im Bereich „Wahlpflichtfächer / Containermodule“ dieses Modulhandbuchs.

Pflichtmodule aller Studienrichtungen

Kryptographie					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	1. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung	Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden kennen die Bedeutung der Schutzziele Datenintegrität, Vertraulichkeit, Authentizität und Verbindlichkeit sowie Methoden und kryptographische Primitiven, die zur Erreichung dieser Schutzziele eingesetzt werden. Sie verstehen die grundlegenden Sicherheitsbegriffe der Kryptographie und können in einfachen Szenarien beurteilen, mit welchen kryptographischen Verfahren und Protokollen nach aktuellem Stand der Technik die Schutzziele erreicht werden können.</p> <p>Sie erwerben die Kompetenz, diese Verfahren und Protokolle unter Zuhilfenahme von Open-Source-Kryptobibliotheken auch selbst zu implementieren.</p> <p>Weiterhin kennen die Studierenden praktisch relevante Angriffe auf die behandelten Schutzmaßnahmen, insbesondere auch solche, die erst durch eine ungeeignete Implementierung möglich werden, und können angemessene Gegenmaßnahmen anwenden.</p>				
	<p>Inhalte</p> <ul style="list-style-type: none"> • Grundlagen <ul style="list-style-type: none"> • Schutzziele der IT-Sicherheit, Angriffsszenarien und Sicherheitsbegriffe • Kryptographische Primitiven <ul style="list-style-type: none"> • historische Verschlüsselungsverfahren und deren Kryptoanalyse • aktuelle symmetrische und asymmetrische Verfahren • Hash-Funktionen • Kryptographische Zufallsbitgeneratoren und deren Beurteilung mittels statistischer Verfahren • Kryptographische Verfahren und Protokolle und deren Anwendungen <ul style="list-style-type: none"> • Message Authentication Codes • Protokolle zur Schlüsselvereinbarung • Hybride Verschlüsselungsverfahren • digitale Unterschriften • Zertifikate, insbesondere deren Verwendung bei TLS • Praktische Angriffe auf kryptographische Verfahren und Protokolle sowie jeweils Gegenmaßnahmen <ul style="list-style-type: none"> • Seitenkanalangriffe • Wörterbuchangriffe • Angriffe auf Grundlage geringer Entropie 				
4	<p>Lehr- und Lernformen</p> <p>Selbststudium in Form von Lehrbriefen, Präsenzlehre als Übung</p>				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: -</p> <p>Inhaltlich: -</p>				

6	Prüfungsformen Mündliche Prüfung
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Steffen Helke, Prof. Dr. Annika Meyer
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): A. Menezes, P. van Oorschot und S. Vanstone: <i>Handbook of Applied Cryptography</i> . CRC Press N. Ferguson, B. Schneier und T. Kohno: <i>Cryptography Engineering - Design Principles and Practical Applications</i> . Wiley Verlag C. Eckert: <i>IT-Sicherheit. Konzepte - Verfahren - Protokolle</i> . Oldenbourg Verlag

IT-Vertragsrecht					
Kennnummer	Workload	Credits nach ECTS	Studien-semester	Häufigkeit des Angebots	Dauer
	150 h	6	1. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden erlernen den praktischen Umgang mit IT-Recht-relevanten Sachverhalten. Dabei machen sie sich mit den Gesetzesgrundlagen und der vertragsrechtlichen Situation im Bereich des IT-Rechts und des Online-Rechts vertraut. Besonders wird dabei auf den für die Wirtschaft relevanten Vertragsbereich des IT-Rechts eingegangen. Die Studierenden lernen, sowohl aus der Perspektive der Anbieter als auch der nachfragenden Wirtschaft die rechtlichen Probleme, Fallstricke und Strukturen zu betrachten.</p> <p>Die Studierenden sind nach Abschluss des Moduls in der Lage, die verschiedenen rechtlichen Möglichkeiten zur Beschaffung und zum Betrieb von Hardware und Software zu unterscheiden, und sie haben die Kompetenz erworben, eine fundierte Entscheidung für die eine oder andere Vorgehensweise zu treffen. Sie sind in der Lage, die wesentlichen Problemfelder solcher Verträge zu erkennen und eigenständig zu verhandeln.</p>				
3	<p>Inhalte</p> <p>Die Veranstaltung orientiert sich in erster Linie an dem Rechts- und Vertragsbedarf der Wirtschaft. Die Erörterungen werden nach den Bedürfnissen der Wirtschaft gegliedert. Immer wiederkehrende Fälle, Probleme und Fallstricken, mit denen sich Unternehmen konfrontiert sehen, werden zum Gegenstand der Vorlesung gemacht und sodann rechtlich erörtert und analysiert.</p> <p>I. IT-Recht</p> <ul style="list-style-type: none"> • Beschaffung von Software, Systemen (Kauf, Leasing, ASP, SAS, Cloudlösungen) • Projektverträge inkl. Projektordnung und Escrow-Verträgen • Pflegeverträge inkl. Service Level Agreement • Schnittstellen zu oben genannten Rechtsbereichen: nämlich <ul style="list-style-type: none"> • Urheberrecht, (insbesondere Lizenzmodelle, GPL, Nutzungsverträge); • Arbeitsrecht, (insbesondere Besonderheiten des Arbeitsrechtes im IT-Recht); • Markenrecht; (Schutz von Software Produkten); • Patentrecht; (Software als Gegenstand des Patentschutzes); <p>II. Onlinerecht</p> <ul style="list-style-type: none"> • Handelsplattformen • Recht der Provider • Domainrecht (insbesondere mit Schnittstellen zum Markenrecht) • Website-Betreuung • Suchmaschinenmarketing – SEM • Fernabsatz, b2b/b2c, inkl. komplexen Plattformlösungen • Social Media (inkl. Schnittstelle Arbeitsrecht, Markenrecht und Allgemeines Zivilrecht) 				

4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen, Präsenzlehre als Übung
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: -
6	Prüfungsformen Klausur
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Andreas Göbel
12	Sonstige Informationen

Funktionale und logische Programmierung					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	2. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden haben die deklarative Programmierung als Alternative zur imperativen Programmierung kennengelernt. Im praktischen Teil haben sie vertiefte Kenntnisse in der funktionalen Sprache Haskell sowie grundlegende Kenntnisse in der logischen Sprache Prolog erworben: Sie können Programme nicht nur lesen und verstehen, sondern auch für eigene Zwecke modifizieren oder vollständig neu entwickeln. Sie können erläutern, warum I/O in deklarativen Programmiersprachen schwierig zu realisieren ist, und beschreiben, wie Haskell und Prolog diese Probleme lösen. Die Studierenden können die Nützlichkeit dieser Programmierparadigmen für eigene Projekte einschätzen und auf Basis eigener praktischer Erfahrungen mit funktionalen und logischen Sprachen begründet entscheiden, ob sie ein Projekt mit klassischen (objektorientierten bzw. prozeduralen) oder mit deklarativen Sprachen umsetzen.				
3	Inhalte <ul style="list-style-type: none"> • Einführung in die funktionale Programmierung mit Haskell: <ul style="list-style-type: none"> • Grundlagen der Sprache Haskell • benutzerdefinierte Typen und I/O • Typklassen und Module • Funktionen höherer Ordnung • Funktor-Erweiterungen und Lambda-Kalkül • Einführung in die logische Programmierung mit Prolog • andere funktionale Sprachen (F#, Scheme); funktional-logische Programmierung (Curry) 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: –				
6	Prüfungsformen Klausur				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): W.F. Clocksin, C.S. Mellish: " <i>Programming in Prolog</i> ". Springer-Verlag M. Lipovača: " <i>Learn You a Haskell for Great Good! A Beginner's Guide</i> ". No Starch Press, San Francisco, CA, USA, 1st edition

NoSQL-Datenbanken					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	2. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden haben unterschiedliche Konzepte nicht-relationaler und verteilter Datenbanken kennengelernt: Sie sind mit den theoretischen Grundlagen vertraut, haben ausgewählte Algorithmen analysiert und können diese auch praktisch anwenden. Sie sind in der Lage, nicht-relationale Datenbanken und die eingesetzten Techniken von einem übergeordneten Standpunkt aus zu betrachten und deren Einsatzmöglichkeiten zu bewerten. Insbesondere können Sie entscheiden und begründen, ob für ein konkretes Projekt klassische relationale oder NoSQL-Datenbanken einzusetzen sind.				
3	Inhalte <ul style="list-style-type: none"> • Grenzen und Erweiterung relationaler Datenbanken • theoretische Grundlagen verteilter Datenbanken (Big Data, Datenfragmentierung, Replikation, Synchronisation und Konsistenz, CAP-Theorem, In-Memory Datenbanken, Lambda- und Kappa Architekturen) • ausgewählte Algorithmen (z.B. Map-Reduce-Algorithmus) • nicht-relationale Datenmodelle und Beispiele von NoSQL-Datenbanksystemen • Polyglot-Persistenz und multimodale Datenbanken 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Das Modul setzt ausgewiesene Kenntnisse in relationalen Datenbanken voraus.				
6	Prüfungsformen Portfolio				
7	Prüfungsvorleistung -				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Uwe Klug
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Edlich/Friedland/Hampe/Brauer/Brückner; <i>NoSQL</i> ; Hanser Verlag Wiese; <i>Advanced Data Management</i> ; De Gruyter Verlag Kleppmann; <i>Designing data-intensive applications</i> ; O'Reilly Media

Master-Konferenzseminar					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen Aufaktveranstaltungen: 4 h Seminar: 4 h		Kontaktzeit 8 h	Selbststudium 142 h	geplante Gruppengröße 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden arbeiten sich selbständig anhand wissenschaftlicher Literatur in ein vorgeschlagenes oder nach Absprache frei gewähltes Thema aus einem von Semester zu Semester wechselnden Teilgebiet der Informatik ein, erstellen dazu eine schriftliche Ausarbeitung, bereiten einen Seminarvortrag vor und halten ihn.</p> <p>Zu den hier zu erwerbenden Kompetenzen zählen die selbständige Literaturrecherche, korrektes Zitieren, die Fähigkeit auszuwählen, welche Aspekte in der Ausarbeitung und im Vortrag behandelt werden, zielgruppengerechtes Schreiben sowie der Umgang mit dem Textsatzsystem LaTeX.</p> <p>Das Seminar findet als „Konferenzseminar“ statt: Die Studierenden üben auch das Prozedere ein, das beim Einreichen einer wissenschaftlichen Arbeit bei einer Konferenz üblich ist: Unter anderem lernen sie, die Arbeiten anderer Personen im Review-Prozess zu beurteilen und Kritik und Verbesserungsempfehlungen auszusprechen sowie diese für die eigene Arbeit anzunehmen, zu bewerten und umzusetzen.</p>				
3	Inhalte <ul style="list-style-type: none"> wissenschaftliches Arbeiten, Zitieren, Plagiat, LaTeX-Grundlagen, Abläufe einer wissenschaftlichen Konferenz aktuelle Themen der Informatik (mit wechselndem Themenschwerpunkt) 				
4	Lehr- und Lernformen Seminar Selbststudium (Recherche und Lektüre wissenschaftlicher Literatur) Präsenzlehre als seminaristischer Unterricht				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: –				
6	Prüfungsformen Portfolio				
7	Prüfungsvorleistung -				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) –				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Balzert, Helmut et al.: <i>Wissenschaftliches Arbeiten: Ethik, Inhalt & Form wiss. Arbeiten, Handwerkszeug, Quellen, Projektmanagement, Präsentation</i> , Springer Nature

Compilerbau					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	4. Semester.	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden haben sich in die Theorie formaler Sprachen eingearbeitet und kennen die Chomsky-Hierarchie. Sie können Typ-3- und Typ-2-Grammatiken angeben und kennen Verfahren, um formale Eigenschaften der Grammatiken zu überprüfen. Für einfache Sprachen können die Studierenden selbstständig Grammatiken definieren und diese verwenden, um Lexer und Parser für die Sprachen zu entwerfen. Sie können die einzelnen Phasen eines Compilers benennen und die zentralen Datenstrukturen und Aufgaben erläutern, die in den jeweiligen Phasen zum Einsatz kommen.				
3	Inhalte <ul style="list-style-type: none"> • Chomsky-Hierarchie • reguläre Sprachen • endliche Automaten • kontextfreie Sprachen und Grammatiken • Entwurf von Scannern und Parsern • Parser-Generatoren • Grundlagen der Berechenbarkeits- und Komplexitätstheorie 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Vorlesung, seminaristischer Unterricht und Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Programmierkenntnisse (C/C++, Java), Unix Shell				
6	Prüfungsformen Klausur				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) –				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Jeffrey D. Ullman, Monica S. Lam, Ravi Sethi und Alfred V. Aho: <i>Compiler: Prinzipien, Techniken und Werkzeuge</i> , Pearson Michael Sipser: <i>Introduction to the Theory of Computation</i> , Cengage Learning Emea John E. Hopcroft, Rajeev Motwani und Jeffrey D. Ullman: <i>Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit</i> , Pearson Uwe Schöning: <i>Theoretische Informatik kurz gefasst</i> , Spektrum Akademischer Verlag

Netzökonomie					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	5. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen die Netzökonomie als jene Teildisziplin der Wirtschaftsinformatik und der Wirtschaftswissenschaften, die den wirtschaftlichen Einsatz computerbasierter Vernetzung zur Kommunikation und Durchführung betriebswirtschaftlicher Transaktionen behandelt. Sie verstehen die aktuellen wissenschaftlichen und informatischen Grundlagen und Methoden digital vernetzter Ökonomien, wie die Theorie von Netzwerken sowie die Analyse von Daten, Zeitreihen und Sozialen Netzwerken, können die wesentlichen Begriffe abgrenzen und neue Entwicklungen einordnen. Durch praktische Übungen vertieft können sie die erlernten Methoden in der betrieblichen Praxis anwenden und einsetzen.				
3	Inhalte <ul style="list-style-type: none"> • Netzwerktheorie: Mathematische Grundlagen (quadratische Matrizen und Eigenwerte), Graphentheorie, Netzwerkstrukturen (Zufallsnetze und skalenfreie Netze, Small-World-Phänomen) • Datenanalyse und maschinelles Lernen: statistische Analyseverfahren, parametrische statistische Modelle, nichtlineare Regression • Zeitreihenanalyse: Autoregressive Modelle mit gleitendem Durchschnitt, nichtstationäre und periodische Prozesse • Soziale Netzwerkanalyse und Netzwerkeffekte: Externalitäten digitaler Märkte, Braess-Paradox, Kaskaden und Viralität, Zentralitäten (Eigenvektorzentralität, Katz-zentralität, PageRank, umkehrfreie Zentralität und Hashimoto-Matrix), Clusteranalysen (Louvain-Methode) 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Programmiererfahrung, Grundkenntnisse der linearen Algebra mit Matrizen und der Statistik				
6	Prüfungsformen Klausur				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) –				

10	Stellenwert der Note für die Endnote 6/120
11	Modulbeauftragte*r und hauptamtlich Lehrende Felix Neubürger
12	Sonstige Informationen Literaturlauswahl (jeweils in der aktuellen Auflage): Backhaus, K.; Erichson, B.; Plinke, W.; Weiber, R.: <i>Multivariate Analysemethoden</i> . Berlin Heidelberg: Springer Gabler Backhaus, K.; Erichson, B.; Weiber, R.: <i>Fortgeschrittene Multivariate Analysemethoden</i> . Berlin Heidelberg: Springer Gabler Bofinger, P.: <i>Grundzüge der Volkswirtschaftslehre. Eine Einführung in die Wissenschaft von Märkten</i> . München: Pearson Studium Géron, A.: <i>Hands-On Machine Learning with Scikit-Learn and Tensor-Flow</i> . Sebastopol: O'Reilly Clement, R.; Schreiber, D.: <i>Internet-Ökonomie. Grundlagen und Fallbeispiele der vernetzten Wirtschaft</i> . Berlin Heidelberg: Springer Gabler Diestel, R.: <i>Graph Theory</i> . Berlin Heidelberg: Springer-Verlag Easley, D.; Kleinberg, J.: <i>Networks, Crowds, and Markets. Reasoning about a Highly Connected World</i> . Cambridge New York: Cambridge University Press Kaderali, F.; Poguntke, W.: <i>Graphen, Algorithmen, Netze. Grundlagen und Anwendungen in der Nachrichtentechnik</i> . Braunschweig Wiesbaden: Vieweg Knieps, G.: <i>Netzökonomie: Grundlagen - Strategien - Wettbewerbspolitik</i> . Wiesbaden: Gabler Verlag Newman, M. E. J.: <i>Networks. An Introduction</i> . Oxford New York: Oxford University Press Ng, A.; Soo, K.: <i>Data Science – was ist das eigentlich?! Algorithmen des maschinellen Lernens verständlich erklärt</i> . Berlin: Springer Shumway, R. H.; Stoffer, D. S.: <i>Time Series Analysis and Its Applications</i> . Cham: Springer Vogel, J.: <i>Prognose von Zeitreihen. Eine Einführung für Wirtschaftswissenschaftler</i> . Wiesbaden: Springer Gabler

Projekt					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	450 h	18	5. Semester	Winter- und Sommersemester	1 Semester
1	Lehrveranstaltungen 4 SWS Seminar	Kontaktzeit 45 h	Selbststudium 405 h	geplante Gruppengröße i.d.R. 1 Studierende*r	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden erweitern und vertiefen die Inhalte einzelner Veranstaltungen. Sie erarbeiten eigene Beiträge zu laufenden Forschungs- und Entwicklungsarbeiten und lösen komplexe Aufgabenstellungen im Team oder in Einzelarbeit. Dabei haben sie selbständiges und forschungsbezogenes Arbeiten erlernt und eingeübt.</p> <p>Projektmanagementkompetenzen erwerben die Studierenden auf fortgeschrittenem Niveau, da für dieses Projekt eine dem Masterstudium angemessene Selbständigkeit notwendig ist und auch im Verlauf des Projekts demonstriert wird.</p> <p>Wird die Projektarbeit im Team ausgeführt, wird auch die Befähigung zur Teamarbeit trainiert und verbessert.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Durchführung von Einzel- oder Gruppenprojekten • Aufgabenstellungen aus den Bereichen der Lehrgegenstände des Master-Studienganges und den laufenden Forschungs- und Entwicklungsarbeiten des Fachbereiches 				
4	<p>Lehr- und Lernformen</p> <p>Bestandteile des Ablaufs können, je nach Art und Ziel der jeweiligen Veranstaltung, folgende Elemente sein:</p> <ul style="list-style-type: none"> • Festlegung der Aufgabenstellung durch Aufgabenbeschreibung, Pflichtenheft und Literaturliste • Auftaktveranstaltung für die Teilnehmer*innen eines Projektes oder alle Teilnehmer*innen aller Projekte des laufenden Semesters • Zwischenpräsentation / Abschlusspräsentation • Schriftliche Ausarbeitung (bei Software-Erstellung zusätzlich Programmquellen, Nutzer- und Entwicklungsdokumentation) 				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: –</p> <p>Inhaltlich: in der Regel Kenntnisse in dem zu vertiefenden Gebiet, die durch den erfolgreichen Besuch der einschlägigen Lehrveranstaltungen nachgewiesen wurden.</p>				
6	<p>Prüfungsformen</p> <p>Klausur, semesterabschließende schriftliche Ausarbeitung oder mündliche Prüfung</p>				
7	<p>Prüfungsvorleistung</p> <p>–</p>				
8	<p>Voraussetzungen für die Vergabe von Kreditpunkten</p> <p>bestandene Modulprüfung</p>				

9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 12/120 in 6-semesteriger bzw. 12/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Dozent*innen der Angewandten Informatik
12	Sonstige Informationen Literatur wird themenspezifisch im Einzelfall festgelegt.

Pflichtmodule Anwendungsentwicklung

Es folgt eine Übersicht der Pflichtmodule, die belegt werden müssen, wenn die Studienrichtung *Anwendungsentwicklung* gewählt wird:

- Modellgetriebene Software-Entwicklung
- Vertiefung Software-Engineering
- Moderne Web-Frameworks
- Usability Engineering

Modellgetriebene Software-Entwicklung					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	1. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden haben einen umfassenden Überblick über die Techniken der Modellgetriebenen Softwareentwicklung (MDSD) erhalten. Sie sind in der Lage, die Techniken der MDSD in eigenen Projekten einzusetzen; konkret können sie eine webbasierte Unternehmenslösung in Java mit Hilfe des Spring Boot Frameworks entwickeln.</p> <p>Sie kennen u. a. verschiedene Diagrammart der UML und können erläutern, wie man Statik und Dynamik von Softwaresystemen mit der UML modellieren kann. Sie können UML-Klassendiagramme anwenden, um die Datenstrukturen innerhalb von Softwaresystemen zu spezifizieren, und sie beherrschen die Methodik, um aus skizzierten bzw. realen Benutzeroberflächen ein Klassendiagramm abzuleiten.</p> <p>Die Studierenden erklären die Prinzipien der Metamodellierung. Sie können MDSD-taugliche Zielplattformen konzipieren und entwerfen, und sie beherrschen wesentliche Techniken der Codegenerierung. Sie erläutern Vorteile domänenspezifischer Sprachen, kennen Werkzeuge zur Entwicklung solcher Sprachen und können diese anwenden.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Modellbegriff • Modellierung mit der UML und der BPMN • Domänenspezifische Sprachen (DSLs) • MDSD-taugliche Zielarchitekturen • Codegeneratoren und Interpreter • MDA, Metadatenarchitekturen (MOF) • Metamodellierung • Testen • aktuelle MDSD-Tools • Produktlinien 				
4	<p>Lehr- und Lernformen</p> <p>Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung</p>				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: –</p> <p>Inhaltlich: Softwaretechnische Grundkenntnisse (UML, Java Standard und Enterprise Edition), wie sie in einem Bachelor- oder Diplomstudiengang der (angewandten) Informatik vermittelt werden.</p>				

6	Prüfungsformen Semesterabschließende schriftliche Ausarbeitung
7	Prüfungsvorleistung -
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) -
10	Stellenwert der Note für die Endnote 6/120
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Doga Arinir
12	Sonstige Informationen Literaturauswahl (in der jeweils aktuellen Auflage): Thomas Stahl, Markus Völter, Sven Efftinge, Arno Haase: <i>Modellgetriebene Softwareentwicklung</i> , dpunkt.verlag

Vertiefung Software-Engineering					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	2. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden sind mit den in großen Softwareentwicklungsprojekten auftretenden Problemen vertraut. Sie kennen den Prozess des Software-Engineerings sowie die Ergebnistypen der einzelnen Prozessschritte und verstehen ihren Einfluss auf den Erfolg eines Softwareentwicklungsprojektes.</p> <p>Sie können bei gegebenen Anforderungen die Teststrategie für ein Softwareprojekt entwickeln und die verschiedenen Teststufen planen.</p> <p>Durch verschiedene Best-Practice-Beispiele haben die Studierenden die theoretischen Ansätze des Software Engineerings in konkreten Implementierungen gesehen und können diese auf neue Situationen übertragen.</p>				
3	Inhalte <p>Es werden zentrale Themen des Software Engineerings aufgegriffen und detailliert bearbeitet. Dabei erhalten die Studierenden einen vertieften Einblick in die Themenbereiche:</p> <ul style="list-style-type: none"> • Konfigurationsmanagement • Softwaremetriken • Softwaretest und Integration • Softwareentwurfsmodelle (funktions-/daten-/objektorientiert) • Architektur- und Entwurfsmuster 				
4	Lehr- und Lernformen <p>Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung</p>				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Das Modul setzt Kenntnisse voraus, wie sie in den Modulen „Java-Programmierung 1“ und „Software-Engineering“ unseres Bachelor-Studiengangs vermittelt werden.				
6	Prüfungsformen <p>Semesterabschließende schriftliche Ausarbeitung</p>				
7	Prüfungsvorleistung <p>-</p>				
8	Voraussetzungen für die Vergabe von Kreditpunkten <p>bestandene Modulprüfung</p>				
9	Verwendung des Moduls (in anderen Studiengängen) <p>-</p>				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Doga Arinir
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): H. Balzert, <i>Lehrbuch der Software Technik</i> ; Bd. 1+2, Spektrum Verlag J. Goll, M. Dausmann; <i>Architektur- und Entwurfsmuster in der Softwaretechnik</i> ; Springer Verlag J. Goll, <i>Methoden des Software Engineering</i> G. Starke; <i>Effektive Software-Architekturen</i> ; Hanser Verlag A. Schatten, S. Biffi, M. Demolsky, E. Gostischa-Franta, Th. Östreicher, D. Winkler; <i>Best Practice Software-Engineering</i> ; Spektrum Akademischer Verlag

Moderne Web-Frameworks					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Praktikum		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden verstehen die Funktionsweise moderner Webanwendungen und kennen die Vor- und Nachteile verschiedener Ansätze wie etwa Rich Internet Applications oder Single-Page-Anwendungen. Sie können erläutern, welche Komponente die Präsentationslogik enthält. Die Studierenden haben insbesondere durch praktische Rechnerübungen sowohl Frontends als auch Backends von Web-Anwendungen auf Basis populärer Frameworks implementiert, und sie sind in der Lage, Web-Anwendungen vergleichbarer Komplexität selbständig zu planen und zu entwickeln.				
3	Inhalte <ul style="list-style-type: none"> • Lehrbriefe <ul style="list-style-type: none"> ○ Ereignisgetriebene Architekturen ○ Single Page Anwendungen ○ Moderne Web-Frameworks wie z. B. Vue.js, React oder Angular • Praktikum <ul style="list-style-type: none"> ○ Bearbeitung von darauf abgestimmten Präsenzaufgaben ○ Erstellung einer einfachen Webanwendung mithilfe verschiedener Frameworks ○ kritische Reflektion unterschiedlichen Ansätze 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhalt: Das Modul setzt die Inhalte der Module „Grundlagen der Informatik“, „Internet-Technologien“ und „Datenbanken“ aus unserem Bachelor-Studiengang voraus.				
6	Prüfungsformen Semesterabschließende schriftliche Ausarbeitung				
7	Prüfungsvorleistung –				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) –				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen

Usability Engineering					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	4. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden werden in die Lage versetzt, effektiv und effizient bei der Gestaltung von interaktiven IT-Systemen mitzuwirken, die das Bedürfnis des Menschen nach einer zufriedenstellenden, abwechslungsreichen, selbst bestimmten und produktiven Nutzung der modernen Informationstechnik befriedigen. Sie verstehen, dass diese so genannte menschenzentrierte Gestaltung von interaktiven IT-Systemen die Teilhabe an der zunehmend digitalisierten Gesellschaft entscheidend fördert.				
3	Inhalte <ul style="list-style-type: none"> • Relevante, psychologische und physiologische Erkenntnisse über den Menschen • Ziele einer menschenzentrierten Gestaltung interaktiver IT-Systeme und erprobte Phasenmodelle für den Gestaltungsprozess • Empirisch gesicherte Regeln, anerkannte Heuristiken, empfohlene Normen und verbindliche Verordnungen für die menschenzentrierte Gestaltung von interaktiven IT-Systemen • Methoden für die menschenzentrierte Gestaltung und Evaluation von interaktiven IT-Systemen • Demonstration und kritische Diskussion vielfältiger Aspekte von Standard-Komponenten einer menschengerechten Informationstechnik • Praktische Entwicklung und Evaluation von konkreten Komponenten interaktiver IT-Systeme 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Das Modul setzt Kenntnisse des Moduls „Softwareengineering“ aus dem Bachelor-Studiengang oder vergleichbare Vorkenntnisse voraus.				
6	Prüfungsformen Mündliche Prüfung				

7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Rainald Schöneberg
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Sven Heinsen, Petra Vogt (Hrsg.): <i>Usability praktisch umsetzen</i> , Carl Hanser Verlag Deutsche Akkreditierungsstelle GmbH (Hrsg.): <i>Leitfaden Usability</i> , https://www.dakks.de/content/leitfaden-usability Deborah Mayhew: <i>The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design</i> , Morgan Kaufmann Jacob Nielsen: <i>Usability Engineering</i> , Morgan Kaufmann Michael Richter, Markus D. Flückiger: <i>Usability und UX kompakt</i> , Springer Vieweg Henning Brau, Florian Sarodnick: <i>Methoden der Usability Evaluation - Wissenschaftliche Grundlagen und praktische Anwendung</i> , Hogrefe AG Heide Balzert, Uwe Klug, Anja Pampuch: <i>Webdesign & Web-Usability</i> , W3L-Verlag

Pflichtmodule Systemintegration

Es folgt eine Übersicht der Pflichtmodule, die belegt werden müssen, wenn die Studienrichtung *Systemintegration* gewählt wird:

- Server-Betriebssysteme
- Technische Dokumentation
- Netzwerksicherheit
- Cloud Computing

Server-Betriebssysteme					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	1. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Praktikum		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden haben die wichtigsten Konzepte der Systemadministration für Serversysteme kennengelernt und können sie in der Praxis anwenden.</p> <p>In verschiedenen Bereichen, in denen zur Problemlösung mehrere Werkzeuge bereitstehen (etwa bei der Virtualisierung), lernen die Studierenden, die Alternativen zu evaluieren und pragmatische Entscheidungen zu treffen.</p> <p>Die Konfiguration diverser Systemdienste haben die Studierenden durch den Einsatz grundlegender Tools (etwa: Nutzung von Kommandozeilenwerkzeuge und Bearbeiten von Konfigurationsdateien) eingeübt.</p> <p>Im Bereich der Serversicherheit sind die Studierenden für verschiedene Fehlerquellen und Angriffsszenarien sensibilisiert, und sie haben Methoden zur Gefahrenabwehr und zum Härten (Hardening) kennengelernt.</p>				
3	Inhalte <p>Anhand konkreter und aktueller Server-Betriebssysteme (entweder Linux oder Microsoft Windows) wird eine Auswahl der folgenden Themen behandelt:</p> <ul style="list-style-type: none"> • Booten und Systeminitialisierung • Kernel, Module und Treiber • Software-Installation, automatisierte Software-Verteilung • Datensicherung, Replikation • Syslog, Log-Files, Monitoring • Systemsicherheit, Hardening • Leistungsmessung und Leistungsbeurteilung • E-Mail-, Web-, Datei- und Druckdienste • Virtualisierung und Containerisierung 				
4	Lehr- und Lernformen <p>Selbststudium in Form von Lehrbriefen</p> <p>Präsenzlehre als Praktikum</p>				
5	Teilnahmevoraussetzungen <p>Formal: –</p> <p>Inhaltlich: Grundkenntnisse der Betriebssystem-Administration, wie sie in den Modulen „Microsoft-Betriebssysteme“ und „Unix-artige Betriebssysteme“ unseres Bachelor-Studiengangs vermittelt werden</p>				

6	Prüfungsformen Klausur
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): D. Deimeke et al.: <i>Linux-Server: das umfassende Handbuch</i> , Rheinwerk Computing K. Rankin: <i>Linux Hardening in Hostile Networks: Server Security from TLS to Tor</i> . Pearson

Technische Dokumentation					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	2. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden erwerben und trainieren Kompetenzen im Bereich der Dokumentation und Wissensvermittlung, womit sie ihre fachlich-technische Qualifikation abrunden, denn neben der Zusammenarbeit und Kommunikation im Team ist regelmäßig auch das Informieren bzw. Berichten an Anwender/innen und leitende Mitarbeiter/innen ohne technische Ausbildung/Kenntnisse erforderlich.</p> <p>An Beispielen und mit branchenüblichen Werkzeugen üben die Studierenden ein, Produkte, Prozesse und Software zielgruppengerecht zu dokumentieren, Schulungsunterlagen zu erstellen und einzusetzen. Konkret erstellt jede*r Teilnehmer*in eine komplexe Dokumentation zu einem selbst gewählten Software- oder Hardwareprodukt und kann dabei die in diesem Modul gewonnenen Kenntnisse einsetzen.</p> <p>Im Bereich des zivilgesellschaftlichen Engagements (Bedeutung der Informatik für die Gesellschaft) haben die Studierenden sich mit der Herausforderung auseinandergesetzt, Abläufe und Techniken der Informatik in für die Allgemeinheit verständlicher Form darzustellen und damit einen Beitrag zur Wissenschaftskommunikation zu leisten.</p>				
3	<p>Inhalte</p> <ul style="list-style-type: none"> • Grundlagen der Wissensvermittlung (Didaktik, Zielgruppengerechtigkeit) • Texte erstellen und redaktionell bearbeiten, technische Autor/innen betreuen • Schulungsmaterial erstellen und einsetzen (Handouts, Videos, Folien) • Werkzeuge und Standards (DocBook, LaTeX) • Programmdokumentation (Javadoc, Doxygen, Literate Programming) • Knowledge Management • Multimediale Inhalte erstellen, bearbeiten und integrieren 				
4	<p>Lehr- und Lernformen</p> <p>Selbststudium in Form von Lehrbriefen</p> <p>Präsenzlehre als Übung</p>				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: –</p> <p>Inhaltlich: –</p>				
6	<p>Prüfungsformen</p> <p>Semesterabschließende schriftliche Ausarbeitung</p>				
7	<p>Prüfungsvorleistung</p> <p>-</p>				

8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Dietrich Juhl: <i>Technische Dokumentation: Praktische Anleitungen und Beispiele</i> , Springer Vieweg Kieran Morgan: <i>Technical Writing Process</i> William Zinsser: <i>On Writing Well: The Classic Guide to Writing Nonfiction</i> , Harper Perennial

Netzwerksicherheit					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung	Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden analysieren die Sicherheit von IP-basierenden Kommunikationsnetzen und erkennen Bedrohungen und Schwachstellen der Netzwerksicherheit. Untersucht werden die Risiken und Gefahren, die mit der Ausnutzung von Schwachstellen in Protokollen und Anwendungen sowie fehlender Sicherheits-Policies in Unternehmen einhergehen. An praktischen Beispielen betrachten die Studierenden den Ablauf und die Strategie von Angriffen und erlangen die Kompetenz, daraus Maßnahmen für den sicheren Betrieb von vernetzten IT-Systemen abzuleiten.</p> <p>Die praktischen Aspekte der IT-Sicherheit und der (Un-)Sicherheit von konkreten Verfahren und Produkten werden von den Studierenden in Laborübungen analysiert und vertieft.</p>				
	<p>Inhalte</p> <ul style="list-style-type: none"> • Einige Grundbegriffe der IT-Sicherheit <ul style="list-style-type: none"> • Schutzziele (Vertraulichkeit, Integrität, Authentizität, Verfügbarkeit) • Bedrohungen, Angriffertypen, Angriffsszenarien, (Un)Sicherheiten • Angriffe auf die Verfügbarkeit von IT-Systemen (Sync-Flooding, DOS, DDOS, Bot-Netze) • Abhören und Manipulation von Kommunikationsbeziehungen (ARP-, IP-, DNS-Spoofing) • Entwicklung und Umsetzung eines Sicherheitskonzepts, der Security-Policies sowie der Security-Strategy für ein Unternehmen • Schutz der Netzwerkinfrastruktur <ul style="list-style-type: none"> • Absichern der aktiven Netzwerkkomponenten • AAA • Sicherheit auf der Schicht 2 • Sicherheit auf der Schicht 3 • Überwachung und Abschwächung von Bedrohungen <ul style="list-style-type: none"> • Die Überwachungs-Strategie • Absichern der Netzwerkkomponenten, • Access Control Lists und Firewall-Architekturen • (Un-)Sicherheit von Anwendungsprotokollen: SMTP, HTTP, FTP, ... • WLAN-Sicherheit: WEP, WPA, WPA2 • Intrusion Detection (IDS) und Intrusion Prevention Systeme (IPS) • Sicherheitsuntersuchungen von Netzwerken durch Penetrationstests (OpenVAS, Metasploit) • Einsatz von VPN-Technologien zur Absicherung der Netzwerkkommunikation <ul style="list-style-type: none"> • PPTP, L2TP, IPSec, SSL/TLS • Grundlagen einer PKI-Infrastruktur 				

4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung
5	Teilnahmevoraussetzungen Formal: - Inhaltlich: Rechnernetze 1, Kryptographie
6	Prüfungsformen Mündliche Prüfung
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) keine
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Michael Rübsam
12	Sonstige Informationen Literaturlauswahl (jeweils in der aktuellen Auflage): Eckert, Claudia; <i>Netzwerksicherheit Konzepte Verfahren Protokolle</i> ; Oldenbourg Wissenschaftsverlag Schmech, Klaus; <i>Kryptografie</i> ; dpunkt-Verlag Schneider, Bruce; <i>Angewandte Kryptographie: Protokolle, Algorithmen und Sourcecode in C</i> ; Pearson Studium Schneider, Bruce; <i>Secrets and Lies</i> ; Wiley Mitnick, Kevin; <i>Die Kunst der Täuschung</i> ; mitp Singh, Simon; <i>Geheime Botschaften</i> ; dtv Barker, Keith; <i>CCNA Security 640-554 Official Cert Guide</i> ; Pearson <u>Weblinks:</u> <ul style="list-style-type: none"> • www.BSI.de • www.rsalabs.com • www.ccc.de

Cloud Computing					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	4. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Praktikum		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Basierend auf Grundkenntnissen der Virtualisierung und von IT-Infrastrukturen können die Studierenden die Konzepte von Cloud-Computing Lösungen erklären sowie deren technische und ökonomische Vorteile benennen. Sie kennen unterschiedliche Cloud-Dienste und können diese eigenständig über Web- und API-Schnittstellen benutzen. Sie haben erfahren, wie problemspezifische Anwendungsarchitekturen entworfen werden können und sind in der Lage dieses Wissen auf neue Problemstellungen zu übertragen. Anhand konkreter Beispiele haben Sie die Architektur von Cloud-Computing Frameworks kennen gelernt und sind in der Lage, diese Frameworks selbstständig in Betrieb zu nehmen.				
3	Inhalte <ul style="list-style-type: none"> • Technologische Grundlagen von Cloud Computing (Virtuelle Maschinen, Netzwerk-Virtualisierung, Container, etc.) • Cloud Service Ebenen • Beispiele für Cloud Dienste • Cloud APIs • Cloud Storage • Hardware-/Software-Architekturen für Cloud Computing Lösungen • Entwurf Cloud-basierter Anwendungen • Microservices • Serverless 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Vorlesung, seminaristischer Unterricht und Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Programmierkenntnisse, UNIX Shell				
6	Prüfungsformen Semesterabschließende schriftliche Ausarbeitung				
7	Prüfungsvorleistung -				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) Verbundstudiengang Master Angewandte Künstliche Intelligenz
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturlauswahl (in der jeweils aktuellen Auflage): Ian Foster und Dennis B. Gannon: <i>Cloud Computing for Science and Engineering</i> , MIT Press Thomas Erl, Robert Cope und Amin Naserpour: <i>Cloud Computing Design Patterns</i> , Prentice Hall Oliver Liebel: <i>Skalierbare Container-Infrastrukturen: Das Handbuch für Administratoren und DevOps-Teams</i> , Rheinwerk Dan C. Marinescu: <i>Cloud Computing: Theory and Practice</i> , Morgan Kaufmann Edouard Bugnion, Jason Nieh, und Dan Tsafir: <i>Hardware and Software Support for Virtualization</i> , Morgan & Claypool Publishers John J. Geewax: <i>Google Cloud Platform in Action</i> , Manning Michael Solberg und Ben Silverman: <i>OpenStack for Architects</i> , Packt Publishing OpenStack Foundation: <i>OpenStack Stein Administrator Guides</i>

Pflichtmodule Künstliche Intelligenz

Es folgt eine Übersicht der Pflichtmodule, die belegt werden müssen, wenn die Studienrichtung *Künstliche Intelligenz* gewählt wird:

- Data Mining
- Machine Learning
- Big Data Processing
- Deep Learning Anwendungen

Hinweis: Bei den Inhalten dieser Module gibt es teilweise Überschneidungen mit Inhalten der Pflichtmodule der Studienrichtung „Künstliche Intelligenz“ im Bachelorstudiengang Angewandte Informatik.

Diese Themen werden im Rahmen der Master-Module nur wiederholt bzw. beschleunigt eingeführt, da Studierende nicht zwingend notwendig die nötigen Vorkenntnisse haben, wie sie unser Bachelorstudiengang (bei Wahl der Studienrichtung KI) vermittelt.

Data Mining					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	1. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden sind mit den statistischen und informationstechnischen Grundlagen des Data Minings vertraut. Sie kennen die wichtigsten Konzepte für die Speicherung verschiedener Arten großer Datenmengen und sind in der Lage, Datensätze zu analysieren und Rohdaten geeignet aufzubereiten.</p> <p>Sie verstehen es, Data-Mining-Anwendungen für spezielle Anwendungsszenarien zu entwerfen und mit modernen Werkzeugen zu implementieren.</p> <p>Neben den technologischen Kenntnissen entwickeln die Studierenden ein elementares Verständnis für rechtliche und ethische Voraussetzungen und Konsequenzen datengetriebener Verfahren.</p>				
3	Inhalte <ul style="list-style-type: none"> • Akquisition und Kuration von Daten unter strukturellen, technischen und rechtlichen Gesichtspunkten • Aufbereitung und Speicherung strukturierter Daten • Visualisierung von Datenbeständen • Deskriptive Statistik • Explorative Datenanalyse • Einführung in Data-Mining-Werkzeuge 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als seminaristischer Unterricht und Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: –				
6	Prüfungsformen Portfolio				
7	Prüfungsvorleistung -				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Benjamin Buchwitz
12	Sonstige Informationen Literaturauswahl (in der jeweils aktuellen Auflage): Jürgen Cleve, Uwe Lämmel: <i>Data Mining</i> . De Gruyter Pang-Ning Tan, Michael Steinbach, Anuj Karpatne und Vipin Kumar: <i>Introduction to Data Mining</i> . 2. Ausgabe, Pearson

Machine Learning					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	2. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & SWS Übung (als Lehrbrief) 1 SWS Praktikum		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden kennen die wichtigen Algorithmen des überwachten Lernens. Sie beherrschen den grundlegenden Ablauf von Machine-Learning-Projekten, d. h., sie können die Daten aufbereiten, ein geeignetes Modell auswählen und es trainieren, und sie können die generierten Ergebnisse qualitativ bewerten.</p> <p>Darüber hinaus sind sie in der Lage, diese Schritte auf einfache, neue Problemstellungen zu übertragen.</p> <p>Die Studierenden sind sich der Komplexität verschiedener Machine-Learning-Algorithmen bewusst, und sie können Performance-Aspekte der Systemarchitektur für reale Anwendungsfälle einschätzen.</p>				
3	Inhalte <ul style="list-style-type: none"> • Übersicht und Kategorisierung von KI-Algorithmen • effiziente numerische Berechnungen mit Python • Lineare Modelle für Regression und Klassifikation • Entscheidungsbäume und Ensemble-Methoden • Feed-forward-Netze • Algorithmen zum Trainieren von Modellen • Software Frameworks für Machine Learning • Systemarchitekturen für das Training und die Ausführung von Machine-Learning-Modellen 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Vorlesung, seminaristischer Unterricht und Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Skriptsprachen (Python)				
6	Prüfungsformen Semesterabschließende schriftliche Ausarbeitung				
7	Prüfungsvorleistung -				

8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturlauswahl (in der jeweils aktuellen Auflage): Aurélien Géron: <i>Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow</i> . O'Reilly Stephen Marsland: <i>Machine Learning: An Algorithmic Perspective</i> , CRC Press Sebastian Raschka, Vahid Mirjalili: <i>Python Machine Learning</i> . Packt Publishing Ian Goodfellow, Yoshua Bengio, Aaron Courville: <i>Deep Learning</i> , MIT Press Christopher Bishop: <i>Pattern Recognition and Machine Learning</i> , Springer

Big Data Processing					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Praktikum		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 15 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden sind in der Lage die wichtigsten Konzepte, Algorithmen und Datenstrukturen zur Speicherung, Analyse und verteilten Verarbeitung von Big Data zu erklären. Sie kennen die wesentlichen Anforderungen für das Management großer strukturierter und unstrukturierter Daten und können für konkrete Datensätze geeignete Datenverwaltungssysteme auswählen. Studierende, die das Modul absolviert haben, können moderne Big Data Frameworks einrichten und verwenden, um für kleine Fallbeispiele eigene Lösungsansätze umzusetzen.				
3	Inhalte <ul style="list-style-type: none"> • Map-Reduce-Programmiermodell • Datenmodelle für strukturierte und unstrukturierte Daten (relational, dokumentenorientiert graphbasiert) • Verteilte Datenbanken und Dateisysteme • Big-Data Frameworks (Hadoop, Spark) • Cluster-Management (z. B. YARN, Mesos, Kubernetes) • Cloudbasierte Lösungen für Big Data • Algorithmen für Big Data Processing • Stream Processing • Anwendungen (z.B. Social Networks, Recommender Systems, Fraud detection) 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Vorlesung, seminaristischer Unterricht und Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Programmierkenntnisse, Unix Shell				
6	Prüfungsformen Portfolio				
7	Prüfungsvorleistung -				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) –
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturlauswahl (jeweils in der aktuellen Auflage): Jonas Freiknecht und Stefan Papp: <i>Big Data in der Praxis: Lösungen mit Hadoop, Spark, HBase und Hive. Daten speichern, aufbereiten, visualisieren</i> , Hanser Daniel Fasel und Andreas Meier (Hrsg.): <i>Big Data: Grundlagen, Systeme und Nutzungspotenziale</i> , Springer Vieweg Tom White: <i>Hadoop: The Definitive Guide</i> , O'Reilly Matei Zaharia, Holden Karau, Andy Konwinski und Patrick Wendell: <i>Learning Spark: Lightning-Fast Big Data Analysis</i> , O'Reilly Vasiliki Kalavri und Fabian Huesk: <i>Stream Processing with Apache Flink</i> , O'Reilly

Deep Learning Anwendungen					
Kennnummer	Workload	Credits	Studien-semester	Häufigkeit des Angebots	Dauer
	150h	6 CP	4. Semester	Sommersemester	1 Semester
1	Lehrveranstaltungen 1 SWS Übung	Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 25 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden verstehen die Funktionsweise aktueller Architekturen tiefer neuronaler Netze und können sie für praktische Anwendungen in Bereichen wie Bilderkennung, Spracherkennung, Spracherzeugung oder Natural Language Processing anwenden. Sie kennen Bibliotheken und Werkzeuge zum Aufbau und Training tiefer neuronaler Netze und können mit ihrer Hilfe tiefe neuronale Netze aufbauen und trainieren. Sie kennen aktuelle Trainingsalgorithmen und können diese anwenden. Sie kennen typische Trainingsprobleme wie Over- und Underfitting und können die Qualität von trainierten Modellen beurteilen.				
3	Inhalte <ul style="list-style-type: none"> • Deep Learning und Künstliche neuronale Netze (KNN) • Trainingsalgorithmen für tiefe neuronale Netze • Bewertungsverfahren für die Modellgüte • Bibliotheken wie PyTorch und TensorFlow • Werkzeuge zur Überwachung von Trainingsläufen • Modell-Repositories wie HuggingFace • Modellarchitekturen für spezielle Anwendungen: <ul style="list-style-type: none"> • ResNeXt (Bilderkennung) • Transformer • BERT (NLP) • Jasper (ASR) 				
4	Lehrformen Selbststudium in Form von Lernbriefen Präsenzlehre als seminaristischer Unterricht und Übung				
5	Teilnahmevoraussetzungen Formal: Machine Learning Inhaltlich: Python-Kenntnisse				
6	Prüfungsformen Semesterabschließende schriftliche Ausarbeitung				
7	Prüfungsvorleistung -				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen)				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Patterson, Josh; Deep learning : a practitioner's approach, O'Reilly 2017 Chollet, Francois; Deep Learning with Python, 2018 Wartala, Ramon; Praxiseinstieg Deep Learning : mit Python, Caffe, TensorFlow und Spark eigene Deep-Learning-Anwendungen erstellen, O'Reilly 2018

Masterarbeit und Kolloquium

Die Masterarbeit ist eine Prüfungsarbeit. Sie soll zeigen, dass der*die Kandidat*in befähigt ist, innerhalb einer vorgegebenen Frist eine Aufgabe aus der angewandten Informatik in der jeweiligen Wahlrichtung selbstständig mit den in der Anwendung erprobten wissenschaftlichen und fachpraktischen Methoden zu bearbeiten und in fachübergreifende Zusammenhänge zu stellen. Die Masterarbeit ist entweder eine eigenständige Untersuchung oder betrachtet ein bekanntes Thema unter neuen Aspekten. Der Textumfang der Masterarbeit beträgt in der Regel etwa 80 Seiten mit jeweils etwa 50 Zeilen.

Masterarbeit					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	500 h	20	5. bzw. 6. Semester	Wintersemester und Sommersemester	1 Semester
1	Lehrveranstaltungen –	Kontaktzeit 50 h	Selbststudium 450 h	geplante Gruppengröße i.d.R. 1 Studierende*r	
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden demonstrieren die eigenständige erfolgreiche Bearbeitung einer praxisrelevanten wissenschaftlich-technischen Fragestellung. Sie haben die Abschlussarbeit methodisch und inhaltlich vorbereitet und erfolgreich abgeschlossen.</p> <p>In der Masterarbeit haben sie die Aufgabenstellung, die Lösungsansätze, die verwendeten Methoden und die produzierten / gemessenen / beobachteten Ergebnisse in nachvollziehbarer Form dokumentiert. Die Studierenden haben dabei auch ihre überfachlichen Kompetenzen trainiert und erweitert.</p>				
3	Inhalte <p>Vorzugsweise anwendungsorientierte und damit berufsfeldorientierte Fragestellungen aus dem Gesamtbereich der im Studium vermittelten Wissensgebiete – nach Möglichkeit in Zusammenarbeit mit Industrieunternehmen.</p>				
4	Lehr- und Lernformen <p>Eigenständige Literaturstudien, eigene experimentelle Arbeiten und Untersuchungen, persönliche Beratung durch den/die beteiligte/n Professor*in</p>				
5	Teilnahmevoraussetzungen <p>mind. 75 ECTS bei 6- bzw. 45 ECTS bei 5-semesteriger Variante</p>				
6	Prüfungsformen <p>Masterarbeit (schriftliche Ausarbeitung)</p>				
7	Prüfungsvorleistung <p>–</p>				
8	Voraussetzungen für die Vergabe von Kreditpunkten <p>bestandene Masterarbeit</p>				
9	Verwendung des Moduls (in anderen Studiengängen) <p>–</p>				
10	Stellenwert der Note für die Endnote <p>20/120</p>				
11	Modulbeauftragte*r und hauptamtlich Lehrende <p>Ein*e betreuende*r Professor*in der Fachhochschule Südwestfalen</p>				
12	Sonstige Informationen				

Kolloquium					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	100 h	4	5. bzw. 6. Semester	Wintersemester und Sommersemester	–
1	Lehrveranstaltungen –	Kontaktzeit 1 h	Selbststudium 99 h	geplante Gruppengröße i.d.R. 1 Studierende*r	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden können eine Problemlösung aus einer wissenschaftlich-technischen Fragestellung in einem Vortrag darstellen. Dazu gehört, dass sie die zu präsentierenden Inhalte auswählen und gliedern können. Sie demonstrieren den souveränen Umgang mit einem Präsentationswerkzeug, und sie sind in der Lage, den eigenen Lösungsansatz im Fachgespräch mit den Prüfer*innen zu verteidigen.				
3	Inhalte –				
4	Lehr- und Lernformen –				
5	Teilnahmevoraussetzungen Formal: Absolvierte Masterarbeit Inhaltlich: -				
6	Prüfungsformen Mündliche Prüfung				
7	Prüfungsvorleistung –				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) –				
10	Stellenwert der Note für die Endnote 4/120				
11	Modulbeauftragte*r und hauptamtlich Lehrende Ein*e betreuende*r Professor*in der Fachhochschule Südwestfalen				
12	Sonstige Informationen				

Wahlpflichtfächer / Containermodule

Auf den folgenden Seiten finden Sie eine Auswahl von Wahlpflichtfächern / Containermodulen; das Angebot kann sich von Semester zu Semester ändern.

Data Science Anwendungsfälle (Container Datenanalyse)					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	4. Semester	im Sommersemester	1 Semester
1	Lehrveranstaltungen 4 SWS Seminar	Kontaktzeit 60 h	Selbststudium 90 h	geplante Gruppengröße 20 Studierende	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Im Rahmen des Moduls Data Science Anwendungsfälle erarbeiten die Studierenden Lösungen für anwendungsorientierte Probleme an der Schnittstelle zwischen Data Science und Informatik. Die Studierenden sind nach Abschluss des Moduls in der Lage spezifische Anforderungen aus der Praxissituation abzuleiten, diese zu formulieren, Lösungen zu skizzieren und in einer geeigneten Umgebung mit fachgerecht und problemadäquat ausgewählten Werkzeugen prototypisch zu implementieren.</p> <p>Die Teilnehmer*innen bringen sich entsprechend ihres gegenwärtigen Studienschwerpunktes in die Gestaltung eines Softwaresystems und der zugehörigen Infrastrukturen mit dem Ziel der automatisierten Datensammlung, Datenvorverarbeitung, Datenanalyse und Ergebniskommunikation im Back- bzw. Frontend ein. Die Konzeption, Implementierung und Umsetzung erfolgt analog zu Praxisprojekten entsprechend der individuellen Fähigkeits- und Interessenschwerpunkte.</p>				
	<p>Inhalte</p> <ul style="list-style-type: none"> • Projektarbeit aus dem Bereich Data Science und Big Data • Umsetzung von praktischen Aufgabenstellungen im Schnittstellenbereich von Data Science und Informatik • Konzeptionierung und Umsetzung von prototypischen Lösungen • Gestaltung und Implementierung von verteilten Softwaresystemen • Anwendung datenanalytischer Verfahren 				
4	<p>Lehr- und Lernformen</p> <p>seminaristischer Unterricht, Projektarbeiten, Gruppenarbeiten</p>				
5	<p>Teilnahmevoraussetzungen</p> <p>Formal: gemäß RPO und FPO</p> <p>Inhaltlich: Kenntnisse in mindestens einer Programmiersprache (z.B. Python, C, C++, R).</p>				
6	<p>Prüfungsformen</p> <p>Portfolio</p>				
7	<p>Prüfungsvorleistung</p> <p>-</p>				
8	<p>Voraussetzungen für die Vergabe von Kreditpunkten</p> <p>Bestandene Modulprüfung</p>				
9	<p>Verwendung des Moduls (in anderen Studiengängen)</p> <p>Das Modul wird in den folgenden Studiengängen verwendet: Data Science</p>				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Benjamin Buchwitz
12	Sonstige Informationen Modul des Studiengangs Data Science (M. Sc.).

Effiziente Algorithmen (Container Algorithmen)					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung	Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende	
2	Lernergebnisse (learning outcomes) / Kompetenzen Lernergebnisse: Die Teilnehmer*innen lernen fortgeschrittene Methoden für den Entwurf und die Analyse effizienter Algorithmen kennen und können diese anwenden. Sie sind in der Lage, effiziente Algorithmen für typische Einsatzszenarien auszuwählen, zu implementieren, und ihre Leistungsfähigkeit abzuschätzen. Kompetenzen: Vertiefung zum algorithmischen Denken, Methodenkompetenz, Analysefähigkeit, Synthesefähigkeit				
	Inhalte <ul style="list-style-type: none"> • Dynamische Programmierung • Greedy Algorithmen • Vertiefung zu Graphalgorithmen (Kürzeste Wege, Flussalgorithmen) • Effiziente zahlentheoretische Algorithmen (mit Anwendungen auf Kryptographie / Anschluss an Modul "Kryptographie") • Approximationsalgorithmen für komplexe Probleme 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Übung Lernteamcoaching (Selbststudium in Kleingruppen à 6 Teilnehmer*innen, Behandlung verbliebener offener Fragen in der Präsenzübung)				
5	Teilnahmevoraussetzungen Formal: keine Inhaltlich: grundlegendes Wissen über einfache Datenstrukturen (wie Stack/Queue, verkettete Liste, Baum, Hashtabelle o.ä.), inkl. deren Eigenschaften und Fähigkeit, diese selbständig zu implementieren; Grundkenntnisse in Analysemethoden für Algorithmen (Laufzeit einfacher Algorithmen exakt und in O-Notation abschätzen können); Basiswissen zu mathematischen Grundlagen der Kryptografie (z.B. aus Modul "Kryptografie")				
6	Prüfungsformen Klausur				

7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) -
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Robert Rettinger
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Cormen, Leiserson, Rivest, Stein, Algorithmen - Eine Einführung, Oldenbourg Verlag, 4. Auflage 2013. (Ausgewählte Abschnitte ab Teil "Fortgeschrittene Entwurfs- und Analysetechniken")

IT-Management (Container Betriebswirtschaft)					
Kennnummer	Workload 150 h	Credits 6	Studien- semester 4. Sem.	Häufigkeit des Angebots Sommersemester	Dauer 1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung	Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende	
2	<p>Lernergebnisse (learning outcomes) / Kompetenzen</p> <p>Die Studierenden können aus der Unternehmensstrategie und den Geschäftsanforderungen sowie aus den Geschäftsprozessen in einem kontinuierlichen Prozess die Anforderungen an eine IT-Organisation bestimmen. Hieraus leiten die Studierenden die Ziele einer IT-Organisation ab und entwickeln die IT-Strategie eines Unternehmens.</p> <p>Die Studierenden besitzen die Kompetenz, aus der IT-Strategie die strategische Planung der IT- Architektur für ein Unternehmen durchzuführen. Sie sind in der Lage, ein auf Kennzahlen basierendes Controllingssystem für ein Unternehmen aufzubauen und können eine IT-Organisation damit strategisch und operativ steuern.</p> <p>Im zweiten Teil der Lehrveranstaltung steht das operative Management der IT-Landschaft sowie die Organisation und Führung der IT-Organisation eines Unternehmens im Vordergrund.</p> <p>Die Studierenden kennen die grundlegenden Strukturen und Prozesse zum operativen Management der IT-Services. Die Studierenden sind in der Lage unterschiedliche Rollen im IT-Service Management einzunehmen und können eine IT-Infrastruktur effektiv und effizient betreiben. In diesem Zusammenhang berücksichtigen Sie auch die Grundlagen des IT-Risikomanagements und des IT-Sicherheitsmanagements. Darüber hinaus wissen die Studierenden wie die IT-Service-Management-Zertifizierung eines Unternehmens durchgeführt werden kann.</p> <p>Die Studierenden sind in der Lage die Beziehungen zwischen Aufgaben, Menschen, Sachmitteln und Informationen in einer IT-Organisation zu organisieren und die anvertrauten Menschen so zu führen, dass ihre Leitungen und Leistungsbereitschaft optimiert wird.</p>				
	<p>Inhalte</p> <ul style="list-style-type: none"> • Analyse und Dokumentation von Geschäftsprozessen • Von der Geschäftsstrategie zur IT-Strategie (Business Alignment) • Enterprise Architecture Management • Wirtschaftlichkeitsbeurteilung von IT-Investitionen, IT Projekten und IT-Einsatz • IT-Controlling (Kennzahlensysteme und Scorecards sowie deren Messgrößen) • IT-Service-Management Frameworks am Beispiel von ITIL und CobiT (inkl. Risiko- und Sicherheitsmanagement) • Prozessreifegrad am Beispiel von CMMI • IT-Service-Management Zertifizierung gemäß ISO/IEC 20000 • IT-Sicherheits-Zertifizierung gemäß ISO/IEC 27001 • Organisation und Führung im IT-Bereich (IT-Governance) 				
4	<p>Lehrformen</p> <p>Selbststudium in Form von Lernbriefen</p> <p>Präsenzlehre als Übung</p>				

5	Teilnahmevoraussetzungen Formal: keine Inhaltlich: keine
6	Prüfungsformen Klausur, semesterabschließende schriftliche Ausarbeitung oder mündliche Prüfung
7	Prüfungsvorleistung -
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) Dieses Modul wird nur zu Anerkennung von Leistungen aus einer vorhergehenden FPO / MPO angeboten!
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Michael Rübsam
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Ausgabe): Tietmeyer, Ernst: Handbuch IT-Management; 2. Auflage; Hanser, 2007 Bernhard, M.G.; Blomer, R.; Bonn, L.: Strategisches IT-Management, Band 1; symposium, 2003 Bernhard, M.G.; Blomer, R.; Bonn, L.: Strategisches IT-Management, Band 2; symposium, 2003 Brenner, Walter; Witte, Christoph: Erfolgsrezepte für CIOs; Hanser, 2007

Microsoft-Server-Betriebssysteme (Container Verteilte Systeme und Betriebssysteme)					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3./4. Semester	nach Bedarf	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden haben einen Kenntnisstand erreicht, der dem des MCSE (Microsoft Certified Solutions Expert) Server Infrastructure, MCSE Desktop Infrastructure und MCSE Private Cloud entspricht. Dieses Wissen ist theoretisch und praktisch, weil sie alle nötigen Arbeitsschritte auf der virtuellen Praktikumsumgebung (VPU) des Fachbereichs eingeübt haben. Die Studierenden sind im Anschluss an die Modulteilnahme ausreichend vorbereitet, um die drei MCSE-Zertifizierungsprüfungen bei Microsoft ablegen.				
3	Inhalte <ul style="list-style-type: none"> • Entwurf, Implementierung und Konfiguration einer erweiterten Server-Infrastruktur • Implementierung einer Desktop-Infrastruktur und einer Desktop-Applikationsumgebung • Planung, Konfiguration und Bereitstellung einer Private Cloud • Überwachung und Betrieb einer Private Cloud mit System Center 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen (Selbstlernphasen unter Verwendung der E-Learning-Plattform „Microsoft IT-Academy“) Präsenzlehre als Übung				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Grundkenntnisse der Betriebssystem-Administration, wie sie in den Modulen „Microsoft-Betriebssysteme“ und „Unix-artige Betriebssysteme“ unseres Bachelor-Studiengangs vermittelt werden.				
6	Prüfungsformen Klausur, semesterabschließende schriftliche Ausarbeitung oder mündliche Prüfung				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				

9	Verwendung des Moduls (in anderen Studiengängen) Dieses Modul wird nur zu Anerkennung von Leistungen aus einer vorhergehenden FPO / MPO angeboten!
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Hans-Georg Eßer
12	Sonstige Informationen

Praktische Anwendungen von Algorithmen (Container Algorithmen)					
Kennnummer	Workload	Credits	Studien-semester	Häufigkeit des Angebots	Dauer
	180h	6 CP	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen Seminar / Blockveranstaltung: 4 SWS / 45 h	Kontaktzeit 4 SWS / 45 h	Selbststudium 135 h	geplante Gruppengröße Vorlesung: - Praktikum: -	
2	Lernergebnisse (learning outcomes) / Kompetenzen Die Studierenden kennen Strategien zur Lösung komplexer algorithmischer Probleme wie dynamische Programmierung oder die Greedy-Heuristik und können sie in der Praxis anwenden. Basis sind dabei konkrete Problemstellungen aus Wettbewerben wie etwa Google Hashcode.				
3	Inhalte Im Seminar erarbeiten die Studierenden in Dreier- oder Vierergruppen Lösungen zu Problemen aus vergangenen Wettbewerben zum Lösen algorithmischer Probleme, etwa Google Hashcode, und stellen ihre Lösungen und die verwendeten Strategien den anderen Studierenden vor. In ihren Gruppen nehmen die Studierenden an einem Wettbewerb wie Google Hashcode teil. Der konkrete Wettbewerb und der Termin wird den Studierenden zu Beginn des Moduls mitgeteilt.				
4	Lehrformen Seminar				
5	Teilnahmevoraussetzungen Formal: keine Inhaltlich: Grundlagen der Informatik				
6	Prüfungsformen Portfolio				
7	Prüfungsvorleistungen keine				
8	Voraussetzungen für die Vergabe von Kreditpunkten Bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) -				
10	Stellenwert der Note für die Endnote 5/180				
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron				
12	Sonstige Informationen Literaturauswahl: Baka, Benjamin <i>Python Data Structures and Algorithms</i> David Esparza Alba: <i>Algorithms: For Competitive Programming</i>				

Web-Mining (Container Datenanalyse)					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3./4. Semester	nach Bedarf	1 Semester
1	Lehrveranstaltungen 3 SWS als Selbststudium 1 SWS Seminar / Übung als Blockveranstaltung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 20 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen Nach der erfolgreichen Bearbeitung des Moduls sind die Studierenden mit den grundlegenden Ansätzen des Text-Minings vertraut und können auf dieser Basis konkrete Analyseanwendungen konzipieren und diese unter Verwendung aktueller Technologien umsetzen. Sie sind somit in der Lage Web-Content-Analysen durchzuführen. Darüber hinaus können sie die Prinzipien des Web-Structure- und des Web-Usage-Mining erläutern und diese in Analyseprojekten anwenden. Die Studierenden können ferner die Ansätze des semantischen Webs wiedergeben, unter Verwendung konkreter Repräsentationssprachen Texte semantisch anreichern und Inferenzmechanismen zur Gewinnung von neuem Wissen auf der Basis existierender Abfragesprache konzipieren und umsetzen.				
3	Inhalte Die im World-Wide-Web gespeicherten Informationen wachsen kontinuierlich an und stellen somit eine unentbehrliche Quelle diverser Analysevorhaben dar. Im Vordergrund steht hierbei die Analyse der gespeicherten Webseiteninhalte (Web-Content-Mining), die vorwiegend aus schwach- oder unstrukturierten Texten bestehen. Zur Analyse des Web-Contents werden u.a. Ansätze des Text-Minings angewendet. Unter dem Schlagwort „semantisches Web“ bestehen darüber hinaus seit einigen Jahren Bestrebungen, die unstrukturierten Daten mit semantischen Informationen anzureichern, so dass diese effizient durch Maschinen verarbeitet werden können. Die Grundlage hierfür stellen Ontologien zur Verfügung, auf deren Basis Repräsentationssprachen definiert werden, mit denen die semantische Anreicherung erfolgt. Aufbauend auf den Repräsentationssprachen existieren Inferenzmechanismen und Abfragesprachen, die aus den gespeicherten Informationen (implizites) Wissen ableiten können. Zusätzlich zum Web-Content-Mining können weitere Informationen aus der Verlinkungsstruktur der HTML-Seiten (Web-Structure-Mining) und aus der Nutzung der Webseiten (Web-Usage-Mining) gewonnen werden. Das Modul führt in die Grundlagen des Text-Minings ein und zeigt deren Anwendung zur Analyse von textuellen Webseiteninhalten. Außerdem werden die Grundlagen des semantischen Webs behandelt und die Prinzipien der Wissensrepräsentation durch Ontologien sowie die dazugehörigen Inferenzmechanismen eingeführt, wobei diese anhand von konkreten Repräsentationssprachen nachvollzogen werden. Es werden Ansätze zum Web-Structure- und Web-Usage-Mining behandelt.				
4	Lehr- und Lernformen seminaristischer Unterricht, Vorlesung, Übung				
5	Teilnahmevoraussetzungen Formal: gem. BPO/MPO/FPO Inhaltlich: keine				
6	Prüfungsformen Portfolioprüfung				
7	Prüfungsvorleistung -				

8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung
9	Verwendung des Moduls (in anderen Studiengängen) Verwendung in folgenden Studiengängen: Data Science.
10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Christian Gawron
12	Sonstige Informationen Modul des Studiengangs Data Science (M. Sc.)

Wissensbasierte Systeme					
Kennnummer	Workload	Credits nach ECTS	Studiensemester	Häufigkeit des Angebots	Dauer
	150 h	6	3. Semester	Wintersemester	1 Semester
1	Lehrveranstaltungen 2 SWS Vorlesung & 1 SWS Übung (als Lehrbrief) 1 SWS Übung		Kontaktzeit 25 h	Selbststudium 125 h	geplante Gruppengröße 30 Studierende
2	Lernergebnisse (learning outcomes) / Kompetenzen <p>Die Studierenden können verschiedene Verfahren der Künstlichen Intelligenz benennen, bei denen durch Menschen modelliertes Wissen und Regeln zur Problemlösung eingesetzt werden. Sie kennen Algorithmen zur uninformierten und heuristischen Suche und können diese einsetzen, um einfache Probleme (z. B. Zwei-Personen-Spiele mit vollkommener Information) zu lösen.</p> <p>Die Studierenden kennen Datenstrukturen, Methoden und Werkzeuge, mit denen Faktenwissen repräsentiert bzw. ausgewertet werden kann. Für konkrete Aufgabenstellungen können Sie Lösungsansätze entwerfen und diese experimentell umsetzen.</p>				
3	Inhalte <ul style="list-style-type: none"> • Problemlösen durch Suche, Heuristiken • Grundlagen der Aussagen- und Prädikatenlogik • Deduktives Schließen • Expertensysteme • Wissensextraktion und Wissensrepräsentation • semantische Netze, Knowledge Graph 				
4	Lehr- und Lernformen Selbststudium in Form von Lehrbriefen Präsenzlehre als Vorlesung, seminaristischer Unterricht und Praktikum				
5	Teilnahmevoraussetzungen Formal: – Inhaltlich: Programmierkenntnisse (C/C++ oder Java)				
6	Prüfungsformen Klausur, Projektarbeit oder Kombinationsprüfung				
7	Prüfungsvorleistung Studienleistung – die genauen Modalitäten werden zu Semesterbeginn kommuniziert				
8	Voraussetzungen für die Vergabe von Kreditpunkten bestandene Modulprüfung				
9	Verwendung des Moduls (in anderen Studiengängen) –				

10	Stellenwert der Note für die Endnote 6/120 in 6-semesteriger bzw. 6/90 in 5-semesteriger Variante
11	Modulbeauftragte*r und hauptamtlich Lehrende Prof. Dr. Heiner Giefers
12	Sonstige Informationen Literaturauswahl (jeweils in der aktuellen Auflage): Stuart Russell und Peter Norvig: <i>Künstliche Intelligenz</i> , Pearson Christoph Beierle und Gabriele Kern-Isberner: <i>Methoden Wissensbasierter Systeme - Grundlagen - Algorithmen - Anwendungen</i> , Springer Vieweg Judea Pearl: <i>Heuristics: Intelligent Search Strategies for Computer Problem Solving</i> , Addison-Wesley Andreas Dengel (Hrsg.): <i>Semantische Technologien: Grundlagen. Konzepte. Anwendungen</i> , Springer